



19 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

12 Off nl ungungsschrift
10 DE 41 34 392 A 1

51 Int. Cl.⁵:
G 06 F 9/38

21 Aktenzeichen: P 41 34 392.1
22 Anmeldetag: 17. 10. 91
43 Offenlegungstag: 23. 4. 92

151

DE 41 34 392 A 1

30 Unionspriorität: 32 33 31
17.10.90 JP P 2-279653

71 Anmelder:
Mitsubishi Denki K.K., Tokio/Tokyo, JP

74 Vertreter:
Tiedtke, H., Dipl.-Ing.; Bühling, G., Dipl.-Chem.;
Kinne, R., Dipl.-Ing.; Pellmann, H., Dipl.-Ing.; Grams,
K., Dipl.-Ing., Pat.-Anwälte, 8000 München

72 Erfinder:
Ando, Hideki, Itami, Hyogo, JP

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Verfahren und Vorrichtung zum Ungültigmachen von Befehlen in Geräten mit Parallelverarbeitung

57 Ein Prozessor vom Superskalaren-Typ enthält einen Befehlsspeicher, eine Abruf-Einrichtung, die gleichzeitig eine Vielzahl von Befehlen aus dem Befehlsspeicher abrufen, funktionelle Einheiten, die entsprechende vorbestimmte Funktionen ausführen, und eine Decodier-Einrichtung, die die abgerufenen Befehle decodiert und die Befehle an die entsprechenden funktionellen Einheiten abgibt, die parallel verarbeitet werden können. Die Decodiereinrichtung enthält einen Decodierer, der bestimmt, ob ein Sprung in den empfangenen Befehlen enthalten ist und ob ein Sprung gemäß dem Sprungbefehl erzeugt wird. Der Decodierer hängt mit der Befehlsabgabe an die funktionelle Einheit an den Befehl eine Schreibverzögerungs-Kennung, die anzeigt, ob der Befehl nach einem Sprungbefehl kommt, und eine Gültigkeits-Kennung an, die anzeigt, ob der Befehl gültig ist. Die funktionelle Einheit enthält eine Ausführungs-Einheit, die einen Befehl ausführt, und eine Rückschreib-Einheit, die den Maschinenzustand gemäß dem Ausführungsergebnis in der Ausführungs-Einheit ändert. Ein Prozessor vom Superskalaren-Typ enthält einen Steuerschaltkreis, der das Ändern des Maschinenzustandes durch die Rückschreib-Einheit unterbindet, wenn ein Sprung gemäß dem Sprungbefehl erzeugt wird. Die Steuerungs-Einrichtung setzt die Rückschreib-Einrichtung in einen Zustand, der die Änderung des Maschinenzustandes verzögert, wenn noch nicht bestimmt ist, ob ein Sprung gemäß dem Sprungbefehl erzeugt ist, und führt die Änderung des ...

DE 41 34 392 A 1

Beschreibung

Die Erfindung bezieht sich auf parallelverarbeitende Geräte und insbesondere auf einen Prozessor vom Superskalar-Typ. Insbesondere bezieht sich die Erfindung auf ein Steuersystem zur Durchführung und Ungültigmachen von Befehlen, die den Pipeline-Prozessor (funktionelle Einheiten) nach einem Sprungbefehl in einem Prozessor von Superskalar-Typ zugeführt werden.

Gemäß dem Fortschritt in der Halbleitertechnik der letzten Jahre wurde die Leistungsfähigkeit eines Mikroprozessors immer größer und seine Rechengeschwindigkeit wurde ebenfalls immer größer. Obwohl die Zugriffszeit eines Halbleiterspeichers ebenfalls immer geringer wurde, kann dessen Geschwindigkeitssteigerung nicht der Geschwindigkeitssteigerung eines Mikroprozessors folgen, so daß der Zugriff auf einen Halbleiterspeicher einen Engpaß bei der Geschwindigkeitssteigerung eines Prozessors bildet. Daher wurde die Leistungsfähigkeit von Mikroprozessoren durch das Durchführen der Parallelverarbeitung gesteigert. Der Superskalar-Prozessor ist eines der Systeme, mit denen eine solche Parallelverarbeitung realisierbar ist. Wie in Fig. 1 dargestellt, ist ein Prozessor vom Superskalar-Typ (im weiteren als Superskalarprozessor bezeichnet) derart aufgebaut, daß ein Scheduler bzw. eine Steuerung 200 (normalerweise in einem Befehlsdecoder vorgesehen) Parallelismen im Befehlsfluß erkennt und Befehle, die parallel zu verarbeiten sind, den parallel angeordneten Pipeline-Prozessoren (funktionelle Einheiten) P1, P2 und P3 zuführt. Es kann gesagt werden, daß ein Superskalar-Prozessor ein Computer (oder Prozessor) ist der Eigenschaften hat, die nachfolgend beschrieben sind.

- 1) Er ruft gleichzeitig eine Vielzahl von Befehlen ab.
- 2) Er beinhaltet eine Vielzahl von funktionellen Einheiten (Pipeline-Prozessoren) und ist in der Lage, gleichzeitig eine Vielzahl von Befehlen durchzuführen.
- 3) Er erkennt unter der Vielzahl der abgerufenen Befehle diejenigen, die gleichzeitig bearbeitet werden können, um sie den entsprechenden funktionellen Einheiten zuzuführen.

Fig. 2 zeigt ein Blockschaltbild, das eine allgemeine Struktur eines Superskalarprozessors darstellt. Bezugnehmend auf Fig. 2 beinhaltet ein Superskalarprozessor eine Vielzahl funktioneller Einheiten 4, 5, 6 und 7, wobei jede eine vorbestimmte Funktion ausübt, eine Befehlsabruf-Einheit (IF) 2, die gleichzeitig eine Vielzahl von Befehlen aus einem Befehlsspeicher 1 abrufen, eine Befehlsdecoder-Einheit 3, die gleichzeitig die von der Befehlsabruf-Einheit 2 abgerufenen Befehle vom Befehlsspeicher 1 empfängt und die Befehle erkennt, die gleichzeitig bearbeitet werden können, um die erkannten Befehle den entsprechenden funktionellen Einheiten zuzuführen, um Operations-Verarbeitungsergebnisse oder dergleichen zu speichern.

Der Befehlsspeicher 1 beinhaltet allgemein einen Cachespeicher und einen externen Hauptspeicher und speichert die für die Programmausführung notwendigen Befehle.

Die Befehlsabruf-Einheit 2 gibt einen Befehlszeiger IP an den Befehlsspeicher 1 ab und ruft entsprechend dem Befehlszeiger IP des Befehlsspeichers 1 gleichzeitig eine Vielzahl von Befehlen ab.

Die Befehlsdecoder-Einheit 3 beinhaltet einen Befehls-

decoder, der ein Pipeline-Prozessor-Folgeregler ist. Der Befehlsdecoder empfängt und decodiert die Vielzahl von Befehlen, die von der Befehlsabruf-Einheit 2 abgerufen wurden. Der Pipelineprozessor-Folgeregler (ein Befehls-Scheduler) identifiziert den Maschinentyp der decodierten Vielzahl von Befehlen, um gleichzeitig Befehle von unterschiedlichen Maschinentypen an die entsprechenden funktionellen Einheiten abzugeben. Die Maschinentypen bezeichnen, in welcher funktionellen Einheit die Befehle zu bearbeiten sind.

Die funktionellen Einheiten 4-7 sind parallel programmiert und führen die empfangenen Befehle als Antwort auf einen Zeittakt aus. In Fig. 2 sind bezugnehmend beispielhaft 4 funktionelle Einheiten dargestellt und es können maximal 4 Befehle parallel verarbeitet werden.

Die funktionellen Einheiten 4 und 5 sind Einheiten für arithmetische Ganzzahl-Operationen, die die Addition einer ganzen Zahl usw. durchführen und jeweils eine Ausführung-Einheit (EX) und eine Schreib-Einheit (eine Maschinenzustands-Änderungs-Einheit; WB) beinhalten. Die Schreib-Einheit (WB) schreibt das Verarbeitungsergebnis eines Befehls, der in der Ausführung-Einheit ausgeführt wurde, in ein Daten-Register (nicht gezeigt).

Die funktionelle Einheit 6 ist eine Einheit, die den Zugriff (Laden und Speichern von Daten) auf den Datenspeicher 8 ausführt und die eine Adressen-Erzeugung-Einheit (ADR), eine Einheit zum Ausführen des Zugriffs auf den Speicher (MEM), und eine Schreib-Einheit (WB) zum Schreiben von Daten in das Daten-Register (nicht gezeigt) beinhaltet.

In der Schreib-Einheit (WB) der funktionellen Einheit 6 werden die aus dem Datenspeicher 8 geladenen Daten oder die Daten die in dem Datenspeicher 8 gespeichert werden sollen in das Register geschrieben oder aus dem Register gelesen.

Die funktionelle Einheit 7 ist eine Einheit zum Ausführen von arithmetischen Fließkomma-Operationen und beinhaltet drei Ausführung-Einheiten (EX2, EX2, EX3) und eine Schreib-Einheit (WB) zum Schreiben von Ausführungsergebnissen in ein Datenregister (nicht gezeigt). Eine Fließkomma-Zahl ist eine Zahl, die durch das Verwenden eines Exponentens und einer Mantisse dargestellt wird, ohne das die Dezimalkommastelle festgelegt ist. Eine arithmetische Fließkomma-Operation ist eine Operation, die Fließkommazahlen verwendet und bei der im Vergleich zu arithmetischen Operationen mit ganzen Zahlen ein weiterer Zahlenbereich zugelassen ist, wogegen diese Operation mehr Verarbeitungszyklen benötigt als die arithmetischen Operationen mit ganzen Zahlen.

Die Befehlsabruf-Einheit 2, die Befehlsdecoder-Einheit 3 und die funktionellen Einheiten 4 bis 7 sind ebenfalls in den Superskalar-Prozessor parallel programmiert, wobei diese Einheiten einander überlappend arbeiten. Demgemäß werden im Falle, daß kein Freiraum in den Pipeline-Prozessoren vorhanden ist, die Daten oder Befehle, die in dem vorangegangenen Zyklus erzeugt wurden, den entsprechenden Einheiten zugeführt. Ein Befehl, der beispielsweise durch die Befehlsdecoder-Einheit 3 decodiert wurde, ist der in dem vorangegangenen Zyklus abgerufene Befehl.

Fig. 3 zeigt schematisch dargestellt die Aufbaustruktur der Befehlsdecoder-Einheit. Bezugnehmend auf Fig. 3 beinhaltet die Befehlsdecoder-Einheit 3 vier parallel liegende Decoder-Schaltkreise D1 bis D4 und eine Pipeline-Ablaufsteuerung SC, die auf die Ergebnisse der

Decodier-Schaltkreise D1 bis D4 anspricht, um die Befehle zu erkennen, die parallel verarbeitbar sind und um die Befehle an die entsprechenden funktionellen Einheiten abzugeben.

Die Decodier-Schaltkreise D1 bis D4 sind derart ausgebildet, daß sie entsprechend den Befehlen M1 bis M4, die gleichzeitig aus dem Befehlsspeicher 1 abgerufen wurden, entsprechende Befehle decodieren, um die Decodiererergebnisse an die Pipeline-Ablaufsteuerung SC weiterzuleiten. Der Befehl FM, der der Befehlsdecodier-Einheit 3 zugeführt wurde, beinhaltet auch Adressen (logische Adressen im Befehlsspeicher, die der Befehlsabruf-Einheit zugeführt wurden) A1 bis A4, die den jeweiligen Befehlen entsprechen.

Wenn in dem abgerufenen Befehl FM ein Sprungbefehl enthalten ist, steuert die Pipeline-Ablaufsteuerung SC eine Erzeugung eines Sprunges gemäß dem Sprungbefehl, in dem in der Befehlsabruf-Einheit eine Sprungzieladresse gesetzt wird, und die Verteilung der Befehle, die in der Abfolge dem Sprungbefehl untergeordnet sind, an die funktionellen Einheiten. Nun wird unter Bezugnahme auf die Fig. 2 und 3 die Operation auf einfache Weise beschrieben.

Die Befehlsdecodier-Einheit 3 führt eine Befehlsabrufanfrage an die Befehlsabruf-Einheit 2 zu. Die Befehlsabruf-Einheit 2 führt als Antwort auf die Befehlsabrufanfrage einen Befehlszeiger IP auf den Befehlsspeicher 1 zu, um entsprechend dem Befehlszeiger IP aus dem Befehlsspeicher 1 eine Vielzahl von Befehlen abzurufen. Die abgerufenen Befehle M1 bis M4 werden gleichzeitig den Decodier-Schaltkreisen D1 bis D4 zugeführt, die in der Befehlsdecodiereinheit 3 enthalten sind. Die Decodier-Schaltkreise D1 bis D4 decodieren gleichzeitig die zugeführte Vielzahl von Befehlen.

Die Pipeline-Ablaufsteuerung SC detektiert die Befehle, die ohne Rechenbetriebsmittel parallel verarbeitet werden können und Datenregister, die miteinander in den Befehlen konkurrieren, die in den Decodierkreisen D1 bis D4 decodiert sind, und gibt die Befehle ab, die in den entsprechenden funktionellen Einheiten parallel verarbeitet werden können.

Die funktionellen Einheiten, an die die Befehle abgegeben wurden, arbeiten mit Parallelverarbeitung gemäß den abgegebenen Befehlen. Die funktionellen Einheiten 4 bis 7 sind für Parallelrechenbetrieb programmiert und die Verarbeitung wird mittels jeder der Ausführungseinheiten und Schreib-Einheiten, die in Fig. 2 dargestellt sind, durchgeführt.

Die Operationen der Befehlsabruf-Einheit 2, der Befehlsdecodier-Einheit 3 und der Befehls-Ausführungseinheit (funktionelle Einheiten 4 bis 7) sind ebenfalls für Parallelrechenbetrieb programmiert, so daß sie vorbestimmte Operationen einander überlappend ausführen.

Wie zuvor beschrieben ist es möglich die Befehle durch Parallelrechenbetrieb der Operationen von jeder der Einheiten der funktionellen Einheiten, und durch das Durchführen von Parallelverarbeitung in einer Vielzahl von funktionellen Einheiten mit hoher Geschwindigkeit auszuführen.

Beispiele eines Prozessors von superskalaren Typ sind (1) S. McGeady, "The i960CA Superscalar Implementation of the 80960 Architecture", Proceedings of 35th COMPCON, IEEE 1990, pp. 232 - 240 und (2) R. D. Groves et al., "An IBM second generation RISC Processor Architecture", Proceedings of 35th COMPCON, IEEE, 1990, pp. 166 - 172 gezeigt.

Der Stand der Technik gemäß (1) offenbart einen Prozessor mit funktionellen Einheiten, REG, MEM und

CTRL, dem es möglich ist drei von vier gleichzeitig abgerufenen Befehlen parallel auszuführen.

Der Stand der Technik gemäß (2) offenbart einen Prozessor, der einen Festkomma-Prozessor, einen Fließkomma-Prozessor, einen Sprungprozessor und eine Steuereinheit enthält und in dem vier Befehle gleichzeitig abgerufen werden und vier Befehle gleichzeitig ausgeführt werden können.

Wie zuvor beschrieben wird in einem Superskalar-Prozessor eine Vielzahl von Befehlen abgerufen und eine Vielzahl von Befehlen gleichzeitig ausgeführt, so daß es möglich ist eine Verarbeitungsgeschwindigkeit zu erreichen, die höher als in üblichen Prozessoren ist.

Wie in der gemäß Fig. 2 dargestellten Anordnungsstruktur ist es in dem Fall, daß vier Befehle (M1 bis M4; siehe Fig. 3), die gleichzeitig abgerufen wurden, parallel in vier funktionellen Einheiten 4 bis 7 ausgeführt werden, beispielsweise möglich, vier Befehle in vier Zeitaktzyklen zu verarbeiten und auszuführen (in dem Fall, in dem die Pipeline-Prozessoren der funktionellen Einheiten 4, 5 und 6 in einem Wartezustand sind, bis die Verarbeitung durch die funktionelle Einheit 7 beendet ist).

Während die Befehlssteuerung (oder die in der Befehlsdecodier-Einheit enthaltene Pipeline-Ablaufsteuerung SC) die Steuerung der Befehle nicht immer gleichzeitig in den funktionellen Einheiten verarbeitet, so daß eine parallele Verarbeitung wirkungsvoll ausgeführt wird.

Fig. 4 zeigt in einem Diagramm ein Beispiel für das Abgeben von Befehlen, die von der Befehlsdecodier-Einheit abgegeben wurden. Die Befehlsabgabe der Befehls-Decodiereinheit wird im folgenden unter Bezugnahme auf Fig. 4 beschrieben.

Zuerst werden die vier abgerufenen Befehle in einem Zyklus 1 decodiert. Die Befehle 2 bis 4 können nicht mit Befehl 1 parallel verarbeitet werden, so daß nur Befehl 1 an die funktionelle Einheit abgegeben wird.

Die Befehle 2 und 3 können gleichzeitig verarbeitet werden, während Befehl 4, beispielsweise wegen einer abhängigen Beziehung, durch die er ein Verarbeitungsergebnis des Befehls 2 oder des Befehls 3 verwendet, nicht mit dem Befehl 2 und/oder den Befehl 3 parallel verarbeitet werden kann. In dem Zyklus 2 werden nur der Befehl 2 und der Befehl 3 abgegeben.

In einem Zyklus 3 wird der verbleibende Befehl 4 abgegeben. In einem Zyklus 4 werden ein Befehl 5 und ein Befehl 6 von vier neu abgerufenen Befehlen als Befehle abgegeben, die gleichzeitig verarbeitet werden können.

Nun ist die Reihenfolge der Befehlsabgabe derart, daß in dem Fall, daß Befehle nicht parallel verarbeitet werden können, ein Befehl mit einer niedrigeren Adresse bevorzugt abgegeben wird.

Es gibt einen Fall, bei dem Befehle nicht gleichzeitig abgegeben werden können, auch wenn keine solche abhängige Beziehung von Daten vorliegt. Dies ist der Fall, wenn ein Sprungbefehl in den abgerufenen Befehlen enthalten ist. Ein Befehl, der dem Sprungbefehl folgt, hat seine Gültigkeit, die in Abhängigkeit davon bestimmt wird, ob ein Sprung gemäß dem Sprungbefehl erzeugt wurde, so daß er nicht abgegeben werden kann, bis ein Zustand gemäß dem Sprungbefehl bestimmt ist. Ob ein Sprung gemäß dem Sprungbefehl erzeugt ist, wird in der Befehlsdecodier-Einrichtung bestimmt. Nachfolgend werden die Fälle betrachtet, bei denen nicht bestimmt werden kann, ob in dem Zyklus, in dem der Sprungbefehl der Befehlsdecodiereinrichtung zugeführt

wurde, ein Sprung erzeugt wurde. Der Fall eines bedingten Sprungbefehles ist einer dieser Fälle. Im folgenden wird ein spezieller Fall beschrieben.

Der Sprungbefehl ist ein Befehl, bei dem ein Sprung erzeugt wird, falls der Inhalt eines Registers 0 ist, und anderenfalls kein Sprung erzeugt wird.

Das Register enthält jedoch keinen korrekten Wert, wenn nicht das Schreiben gemäß dem anderen vorangegangenen Befehl beendet ist.

In einem solchen Fall ist es notwendig, das die Ausführung des Sprungbefehls so lange aufzuschieben, bis das Schreiben in das Register gemäß dem vorangegangenen Befehl beendet ist.

Spezielle Befehle, die im folgenden beschrieben sind, sind als solche Befehle zu betrachten.

- 1) LOAD R1, 50 (R2)
- 2) BRZ R1, Label
- 3) ADD R4, R5, R6
- 4) SUB R7, R8, R9.

Der Befehl (1) ist ein Befehl, bei dem im Datenspeicher 8 zu den Daten, deren Adressen Inhalt des Registers R2 sind, 50 addiert wird und in das Register R1 geladen werden.

Der Befehl (2) ist ein Befehl, bei dem ein Sprung zu der Marke bzw. "Label" erzeugt wird, wenn der Inhalt des Registers R1 0 ist.

Der Befehl (3) ist ein Befehl, bei dem der Inhalt eines Registers R6 zu dem Inhalt eines Registers R5 addiert wird und das Ergebnis der Addition in das Register R4 geschrieben wird.

Der Befehl (4) ist ein Befehl, bei dem der Inhalt eines Registers R9 von dem Inhalt eines Registers R8 subtrahiert wird und das Ergebnis der Operation in ein Register R7 geschrieben wird.

In dem Fall, daß diese vier Befehle der Befehls-Decodiereinrichtung zugeführt werden, kann der Sprungbefehl BRZ des Befehls (2), bis dahin, daß die Daten gemäß dem Befehl LOAD des Befehls (1) in das Register R1 geschrieben sind nicht bestimmen, ob ein Sprung erzeugt wird.

Die Gültigkeit der Befehle (3) und (4) wird in Abhängigkeit davon bestimmt, ob ein Sprung gemäß dem Sprungbefehl BRZ des Befehls (2) erzeugt ist. Insbesondere dann, wenn bestimmt ist, daß ein Sprung gemäß Befehl (2) erzeugt wurde, sind die Befehle (3) und (4) nicht auszuführen, jedoch sind sie gültig. Falls andererseits, bestimmt ist, daß kein Sprung gemäß Befehl (2) erzeugt wurde, sind die Befehle (3) und (4) gültig und sind den funktionellen Einheiten zur Ausführung zuzuführen. Ein Befehl-Abgabeverfahren, das in diesem Fall bearbeitet werden kann, ist in Fig. 5 dargestellt.

Fig. 5 zeigt in einem Diagramm Befehlsabgabebedingungen für den Fall, daß in den abgerufenen Befehlen ein Sprungbefehl enthalten ist. Im folgenden wird unter Bezugnahme auf Fig. 5 die Befehlsabgabe für den Fall beschrieben, daß der zuvor beschriebene Sprungbefehl BRZ enthalten ist.

In einem Zyklus 0 werden die Befehle (1) bis (4) zum Zuführen zu der Befehlsdecodier-Einheit 3 abgerufen.

In einem Zyklus 1 werden die Befehle (1) bis (4) decodiert.

In einem Zyklus 2 wird der Befehl (1) an eine funktionelle Einheit (die in Fig. 2 gezeigte funktionelle Einheit 6) abgegeben und ausgeführt.

Insbesondere im Zyklus 2 wird eine Adresse des Inhalts des Registers R2 zuzüglich 50 erzeugt. Zu diesem

Zeitpunkt werden die Befehle (2) bis (4) nicht an die funktionellen Einheiten abgegeben und in der Befehls-decodier-Einheit 3 gehalten. In Fig. 5, bezeichnet ID einen Halte-(Warte-)Zustand eines jeden Befehls in der Befehlsdecodier-Einheit.

In einem Zyklus 3 wird gemäß dem Befehl (1) ein Zugriff auf den Datenspeicher 8 durchgeführt. Der Inhalt des Registers R1 ist zu diesem Zeitpunkt noch nicht bestimmt, so daß die Befehle (2) bis (4) in der Befehls-decodier-Einheit 3 gehalten werden.

In einem Zyklus 4 wird gemäß dem Befehl (1) das Einschreiben in das Daten-Register R1 durchgeführt und der Inhalt des Registers R1 bestimmt.

Ob ein Sprung gemäß dem Befehl (2) erzeugt wird, wird durch dieses Datenschreiben bestimmt und es wird bestimmt, daß kein Sprung zu erzeugen ist.

In einem Zyklus 5 werden die Befehle (3) und (4) an die funktionellen Einheiten (4, 5) abgegeben und ausgeführt.

In einem Zyklus 6 werden die Ergebnisse der Befehle (3) und (4) in das Daten-Register geschrieben.

In dem Fall, daß der Inhalt des Datenregisters R1 bestimmt ist und bestimmt ist, daß gemäß Befehl (2) im Zyklus 4 ein Sprung zu erzeugen ist, werden die Befehle (3) und (4) nicht abgegeben, diese Befehle werden ungültig gemacht und eine Operation zum Abrufen eines Sprungziel-Befehles wird im Zyklus 5 ausgeführt.

Bei dem zuvor beschriebenen Verfahren zur Befehlsabgabe erwächst das Problem, daß die Befehlsabgabe an die funktionellen Einheiten so lang gestoppt ist, bis bestimmt ist, ob ein Sprung erzeugt wird, auch in dem Fall, daß kein Sprung erzeugt wird, so daß das Problem erwächst, daß in der Pipeline-Verarbeitung eine Lücke entsteht und die Befehle nicht mit hoher Geschwindigkeit ausgeführt werden können.

Daher ist es Aufgabe der Erfindung ein Gerät zur Parallelverarbeitung zu schaffen, das es ermöglicht, Befehle mit hoher Geschwindigkeit auszuführen, auch in dem Fall, daß ein Sprungbefehl enthalten ist.

Weiterhin ist es Aufgabe der Erfindung einen Prozessor vom Superskalar-Typ vorzusehen, der es ermöglicht, ohne Berücksichtigung eines Sprungbefehls Befehle mit hoher Geschwindigkeit auszuführen.

Weiterhin ist es Aufgabe der Erfindung ein Verfahren zur Befehlsausführung vorzusehen, gemäß dem Befehle, auch in dem Fall, daß ein Sprungbefehl enthalten ist, mit hoher Geschwindigkeit ausgeführt werden können.

Eine weitere Aufgabe ist es, ein Verfahren zur Befehlsausführung in einem Prozessor vom Superskalar-Typ vorzusehen, in dem Befehle, auch wenn ein Sprungbefehl enthalten ist, mit hoher, Geschwindigkeit ausgeführt werden können.

Eine erfindungsgemäße Gerät zur Parallelverarbeitung enthält eine Vorrichtung mit der einem Befehl eine Gültigkeits-Kennung, die anzeigt, ob ein Befehl gültig oder ungültig ist, und eine Schreibverzögerungs-Kennung, die anzeigt, ob ein Befehl darauf wartet oder nicht, daß ein Sprung gemäß eines Sprungbefehls bestimmt wird, hinzugefügt wird, und die die Befehle an eine entsprechende funktionelle Einheit abgibt, eine Steuereinrichtung, die in Abhängigkeit von der Gültigkeit-Kennung und der Schreib-Verzögerungs-Kennung die Operation einer Maschinenzustands-Änderungs-Einheit, die in der funktionellen Einheit enthalten ist, in einen Datenhaltzustand, in einen die Daten ungültig machenden Zustand oder in einem Datenschreibausführungszustand versetzt, und eine Vorrichtung, die als Antwort auf die Erzeugung einer Signalerkennung eines Sprunges

gemäß eines Sprungbefehls den Wert der Gültigkeits-Kennung und der Schreibverzögerungs-Kennung ändert.

Die Gültigkeits-Kennung zeigt an, ob ein entsprechender Befehl gültig oder ungültig ist und ob der Befehl auszuführen ist oder nicht. Die Schreibverzögerungs-Kennung zeigt an, ob das Schreiben des Ergebnisses einer Befehlsausführung in das Datenregister zu verzögern ist.

Eine funktionelle Einheit enthält eine Befehlsausführungseinheit und eine Maschinenzustands-Änderungseinheit, mit denen Daten aus der Befehlsausführungseinheit in das Register geschrieben werden. In dem Fall, daß die Schreibverzögerungs-Kennung einen Schreibverzögerungs-Zustand anzeigt, hält die funktionelle Einheit die Daten aus der Befehlsausführungseinheit fest und verzögert das Schreiben in das Register. Wenn die Gültigkeits-Kennung anzeigt, daß der Befehl gültig ist, führt die Befehlsausführungseinheit einen Befehl von der Abruf/Decodier-Einrichtung aus. Die Operation der Maschinenzustands-Änderungseinrichtung wird als Antwort auf den Wert der Gültigkeits-Kennung und der Schreibverzögerungs-Kennung durchgeführt. Das Datenschieben in das Datenregister wird für einen Befehl nach einem Sprungbefehl gemäß der Schreibverzögerungs-Kennung aufgeschoben. Falls die Gültigkeits-Kennung anzeigt, daß der Befehl gültig ist, und auch die Erzeugung des Sprunges gemäß einem Sprungbefehl erkannt ist, wird der Wert der Schreibverzögerungs-Kennung so verändert, daß der Wert anzeigt, daß das Schreiben in das Register nicht auszuführen ist. Falls kein Sprung gemäß einem Sprungbefehl erzeugt wird und angezeigt ist, daß der Haltebefehl auszuführen ist, wird das Schreiben in das Datenregister sofort ausgeführt.

Demgemäß ist es nicht notwendig die Abgabe eines Befehls nach einem Sprungbefehl in eine funktionelle Einheit zu verzögern, bis bestimmt ist, ob ein Sprung gemäß dem Sprungbefehl erzeugt ist, und es ist eine Befehlsverarbeitung mit hoher Geschwindigkeit möglich.

Nachfolgend wird die Erfindung unter Bezugnahme auf die Zeichnung anhand von Ausführungsbeispielen beschrieben.

Es zeigen:

Fig. 1 ein Diagramm, das ein Strukturkonzept eines parallel verarbeitenden Gerätes darstellt,

Fig. 2 ein Diagramm, das die allgemeine Struktur eines Superskalar-Prozessors darstellt,

Fig. 3 ein Diagramm, das die Struktur einer Befehls-Decodiereinheit in einem Superskalar-Prozessor darstellt,

Fig. 4 ein Diagramm, das die Abgabebedingungen von Befehlen der Befehls-Decodiereinheit an die funktionellen Einheiten in einem Superskalar-Prozessor darstellt,

Fig. 5 ein Diagramm, das die Befehls-Abgabe/Ausführungsbedingungen zur Sprungerzeugung in üblichen parallel verarbeitenden Geräten darstellt,

Fig. 6 ein Diagramm, das schematisch ein erfindungsgemäßes Befehls-Ausführungssystem darstellt,

Fig. 7 ein logisches Diagramm, das schematisch die Struktur einer funktionellen Einheit in einem erfindungsgemäßen parallel verarbeitenden Gerät darstellt,

Fig. 8 ein Kurvendiagramm, das ein Zeittaktsignal, das in dem erfindungsgemäßen parallel verarbeitenden Gerät verwendet wird,

Fig. 9 ein logisches Diagramm, das schematisch die

Struktur einer Register-Schreib-Einheit (Maschinenzustands-Änderungseinheit) gemäß einem Ausführungsbeispiel des erfindungsgemäßen parallel verarbeitenden Gerätes darstellt,

Fig. 10 ein Diagramm, das tabellarisch die Logik darstellt,

Fig. 11 ein Diagramm, das tabellarisch eine mittels einer in Fig. 9 gezeigten "in-logic 1" eingeführte Logik darstellt,

Fig. 12 ein Diagramm, das tabellarisch eine mittels einer in Fig. 9 gezeigten "in logic 2" eingeführte Logik darstellt,

Fig. 13 ein Kurvendiagramm, das eine Funktion der Maschinenzustands-Änderungseinheit für den Fall darstellt, daß bestimmt ist, daß in einem erfindungsgemäßen parallel verarbeitenden Gerät gemäß einem Ausführungsbeispiel ein Sprung erzeugt ist,

Fig. 14 ein Signalkurven-Diagramm, das die Funktion der Maschinen-Änderungseinheit für den Fall darstellt, daß bestimmt ist, daß kein Sprung gemäß einem Sprungbefehl in einem erfindungsgemäßen parallel verarbeitenden Gerät gemäß einem Ausführungsbeispiel erzeugt ist,

Fig. 15 ein logisches Diagramm, das schematisch eine Struktur der Befehls-Einheit einer funktionellen Einheit eines erfindungsgemäßen parallel verarbeitenden Gerätes gemäß einem Ausführungsbeispiel darstellt,

Fig. 16 ein Diagramm, das tabellarisch die Logik der in Fig. 15 dargestellten "In-logic 3" darstellt, und

Fig. 17 ein Diagramm, das tabellarisch den logischen Betrieb des in Fig. 15 dargestellten L-Signalspeicher-Schaltkreises zum Ausgeben einer Gültigkeits-Kennung darstellt.

Zunächst wird vor der Beschreibung eines Ausführungsbeispiels, eine Befehlsabgabe- und Ausführungsabfolge in erfindungsgemäßen parallel verarbeitenden Geräten beschrieben.

Fig. 6 zeigt ein Diagramm, das die Befehlsabgabe- und Ausführungsabfolge in dem erfindungsgemäßen parallel verarbeitenden Gerät darstellt. Die Befehlsabgabe- und Ausführungsabfolge wird in folgendem unter Bezugnahme auf Fig. 6 beschrieben.

Befehle (1) bis (4) sind, wie in Fig. 6 dargestellt, dieselben wie die in Fig. 5 dargestellten Befehle (1) bis (4).

In einem Zyklus 0 werden die Befehle (1) bis (4) abgerufen.

In einem Zyklus 1 werden die Befehle (1) bis (4) in einer Befehls-Decodier-Einheit decodiert.

In einem Zyklus 2 werden die Befehle (1), (3) und (4) an funktionelle Einheiten abgegeben. Gemäß Befehl (1) wird eine Adresse eines Datenspeichers (8) erzeugt. Andererseits werden die Befehle (3) und (4) in den funktionellen Einheiten 4 und 5 ausgeführt, die beispielsweise Einheiten für Ganzzahloperationen sind. Befehl 2 wird in der Befehls-Decodier-Einheit 3 gehalten und es noch nicht bestimmt ist, ob ein Sprung in der Befehls-Decodier-Einheit 1D erzeugt wird.

In einem Zyklus 3 wird ein Zugriff auf den Datenspeicher 8 gemäß Befehl (1) durchgeführt und ein gewünschter Datenwert ausgelesen. Andererseits werden die Ausführungsergebnisse der Befehle (3) und (4) zu einer Maschinenzustands-Änderungseinheit (WB) übermittelt, damit sie dort festgehalten werden. Zu diesem Zeitpunkt ist das Schreiben der Ausführungsergebnisse der Befehle (3) und (4) in die Datenregister noch nicht durchgeführt.

In einem Zyklus 4 wird der aus dem Datenspeicher 8 gelesene Datenwert gemäß Befehl (1) in ein Datenregi-

ster R1 eingeschrieben und damit der Inhalt des Registers R1 bestimmt. Es wird bestimmt, ob ein Sprung gemäß dem Befehl (2) in der Befehlsdecodier-Einheit gemäß der Bestimmung des Inhaltes des Registers R1 zu erzeugen ist. In dem Fall, daß bestimmt ist, daß kein Sprung erzeugt wird, werden die in der Maschinenzustands-Änderungs-Einheit WB festgehaltenen Ausführungsergebnisse der Befehle (3) und (4) gemäß der Bestimmung eines "Nicht-Sprunges" gemäß Befehl (2) in die entsprechende Register R4 und R7 eingeschrieben.

In dem Zyklus 4 werden die in der Maschinenzustands-Änderungs-Einheit WB festgehaltenen Daten der Befehle (3) und (4) für den Fall ungültig gemacht, daß ein Sprung gemäß Befehl (2) bestimmt ist und daß bestimmt ist, daß der Sprung nach "label" erfolgen soll, und die Befehle, die sich in einer Befehls-Ausführungseinheit EX befinden, werden ebenfalls ungültig gemacht (was unter Anwendung einer Gültigkeits-Kennung ausgeführt wird). In dem nächsten Zyklus 5 wird mit der Erzeugung eines Sprunges ein Befehl für das Sprungziel "label" abgerufen.

Ein Vergleich zwischen der in Fig. 6 dargestellten Befehls-Abgabe/Ausführungsabfolge und der in Fig. 5 dargestellten Befehls-Abgabe/Ausführungsabfolge zeigt deutlich, daß erfindungsgemäß die Befehle (3) und (4) zwei Zyklen früher in der Befehls-Abgabe/Ausführungsabfolge ausgeführt werden und daß die Befehlsverarbeitung mit höherer Geschwindigkeit ermöglicht ist. Die erfindungsgemäße Befehls-Abgabe/Ausführungsabfolge wird im folgenden zusammenfassend beschrieben.

- 1) Befehle, denen ein Sprungbefehl nachfolgt, werden an entsprechende funktionelle Einheiten abgegeben und ausgeführt, bevor bestimmt ist, ob gemäß dem Sprungbefehl ein Sprung erzeugt wird.
- 2) Wenn diese nachfolgenden Befehle in einen Zustand des Neuschreibens des Maschinenzustandes gelangen, wird das Neuschreiben des Maschinenzustandes unterbrochen und in einen Wartezustand gebracht.
- 3) Falls bestimmt ist, daß ein Sprung gemäß einem Sprungbefehl erzeugt ist, daß heißt, in dem Fall, daß bestimmt ist, daß ein Sprung erzeugt werden soll, werden die nachfolgenden Befehle ungültig. Falls andererseits bestimmt ist, daß kein Sprung gemäß dem Sprungbefehl erzeugt wird, ist das Neuschreiben des Maschinenzustandes gemäß diesen nachfolgenden Befehlen erlaubt.

Das Neuschreiben des Maschinenzustandes zeigt das Ändern des Zustands einer entsprechenden Einheit gemäß der Ausführung von Befehlen wie beispielsweise zum Neuschreiben des Inhaltes des Daten-Registers, zum Neuschreiben des Inhaltes des Datenspeichers 8 und zum Neuschreiben des Zustands-Kennung, an. Als nächstes wird eine Struktur zur Realisierung der in Fig. 6 dargestellten Befehls-Abruf/Ausführungsabfolge beschrieben.

Fig. 7 zeigt ein Diagramm, das eine Struktur eines Hauptteiles eines erfindungsgemäßen parallel verarbeitenden Gerätes gemäß einem Ausführungsbeispiel darstellt und das schematisch die Struktur eines Teiles der funktionellen Einheit (4 oder 5), zur Durchführung arithmetischer Ganzzahloperationen darstellt.

Gemäß Fig. 7 enthält die funktionelle Einheit (arithmetische Ganzzahloperations-Einheit) eine EX-Einheit um mit einem Rechen- und Steuerwerk (ALU) 16 eine

arithmetischen Operation durchzuführen, und eine WB-Einheit zum Einschreiben des Ausführungsergebnisses in eine Register-Datei 9. Die EX-Einheit stellt in der funktionellen Einheit eine Befehls-Ausführungseinheit dar und die WB-Einheit stellt in der funktionellen Einheit eine Maschinenzustands-Änderungs-Einheit dar.

Die Befehls-Decodiereinheit (ID-Einheit) enthält einen Befehls-Decodierer 10, der eine Vielzahl aus dem Befehlsspeicher (in Fig. 7 nicht gezeigt) abgerufenen Befehle decodiert. Der Befehls-Decodierer 10 enthält einen Decodier-Schaltkreis und eine Pipeline-Abfolgesteuerung, die in Fig. 3 dargestellt ist. Der Befehls-Decodierer 10 decodiert empfangene Befehle, erkennt, ob in diesen empfangenen Befehlen ein Sprungbefehl enthalten ist, und erkennt in dem Fall, daß ein Sprungbefehl enthalten ist, ob die Befehle nach dem Sprungbefehl kommen. Das Erkennen, ob die Befehle nach dem Sprungbefehl kommen oder nicht, wird gemäß den dem Befehls-Decodierer 10 zugeführten Adressen der Befehle ausgeführt, die mit den Befehlen (siehe Fig. 3) verbunden sind.

Der Befehls-Decodierer 10 gibt die decodierten Befehle an die EX-Einheiten der entsprechenden funktionellen Einheiten ab. Zu diesem Zeitpunkt erzeugt der Befehls-Decodierer 10 mit einem Befehl IR eine Kennung inst avail, die anzeigt, ob der Befehl IR gültig ist, und eine Kennung delay inst, die anzeigt, ob der Befehl IR sich hinter einem Sprungbefehl befindet. Die Kennung inst avail ist eine Gültigkeits-Kennung und die Kennung delay inst ist eine Schreibverzögerungs-Kennung, die anzeigt, ob eine Maschinenzustandsänderung gemäß dem Befehl zu verzögern ist.

Die Register-Datei 9 enthält eine Vielzahl von Registern und in jedem Register ist ein vorbestimmter Datenwert gespeichert. Die Struktur, in der die Register-Datei 9 als Datenregister eingesetzt ist, wird beispielsweise in Computern mit reduziertem Befehlssatz (RISC) verwendet. In einem RISC-Gerät erfolgt der Zugriff auf den Datenspeicher 8 nur durch die Befehle zum Laden (LOAD) und zum Speichern (STORE) und alle anderen Verarbeitungen erfolgen durch die Verwendung der Register-Datei 9. Die in Fig. 7 dargestellte Struktur der Register-Datei 9 ist dieselbe wie in RISC-Geräten.

Das in Fig. 7 dargestellte parallel verarbeitende Gerät arbeitet, wie in Fig. 8 dargestellt, als Antwort auf nicht überlappende zweiphasige Zeittaktsignale T und L. Der Beginn eines jeden Zyklus ist durch das Zeittaktsignal T bestimmt.

Die EX-Einheit enthält einen T-Zwischenspeicher 11, der als Antwort auf das Zeittaktsignal T den Befehl IR aus dem Befehlsdecodierer 10 festhält, einen T-Zwischenspeicher 12, der als Antwort auf das Zeittaktsignal T eine Gültigkeits-Kennung inst avail aus dem Befehlsdecodierer 10 festhält, einen T-Zwischenspeicher 13, der als Antwort auf das Zeittaktsignal T eine Schreibverzögerungs-Kennung delay inst aus dem Befehls-Decodierer 10 zwischenspeichert, und T-Zwischenspeicher 14 und 15, die als Antwort auf das Zeittaktsignal T gleichzeitig zwei aus der Registerdatei 9 ausgelesenen Daten, data 1 und data 2, speichern. Die entsprechenden arithmetischen Ganzzahl-Recheneinheiten ausgebildete Register-Datei 9 kann in einem Zyklus das Lesen von zwei Daten und das Schreiben von einem Datenwerk durchführen.

Die EX-Einheit enthält weiterhin L-Zwischenspeicher 17, 18 und 19, die Befehle und Kennungen in entsprechenden T-Zwischenspeicher 11 bis 13 als Antwort auf das Zeittaktsignal L halten. Der L-Zwischenspeicher

17 speichert einen Befehl EX IR als Antwort auf das Zeittaktsignal L. Der L-Zwischenspeicher 18 speichert als Antwort auf das Zeittaktsignal L eine Gültigkeits-Kennung `Ex_inst_avail`, die in dem T-Zwischenspeicher 12 gespeichert ist. Der L-Zwischenspeicher 19 speichert als Antwort auf das Zeittaktsignal L eine Schreibverzögerungs-Kennung `Ex_delay_inst`, die in dem T-Zwischenspeicher-Schaltkreis 13 gespeichert ist. Das Zeichen "EX", vor den Signalen und Kennungen zeigt an, daß sie an die Ausführungs-Einheit abgegeben wurden.

Die EX-Einheit enthält weiterhin einen L-Zwischenspeicher 20, der als Antwort auf das Zeittaktsignal L ein Ausgangssignal des Rechen- und Steuerwerkes (ALU) 16 festhält.

Die WB-Einheit enthält einen T-Zwischenspeicher 21, der als Antwort auf das Zeittaktsignal T einen Befehl des L-Zwischenspeicher 17 hält, einen T-Zwischenspeicher-Schaltkreis 22, der als Antwort auf das Zeittaktsignal T die in dem L-Zwischenspeicher 18 gespeicherte Gültigkeits-Kennung `Ex_inst_avail` L festhält, einen T-Zwischenspeicher 23, der als Antwort auf das Zeittaktsignal T die in dem L-Zwischenspeicher 19 gespeicherte Schreibverzögerungs-Kennung `Ex_delay_inst` L festhält, und einen T-Zwischenspeicher 24, der als Antwort auf das Zeittaktsignal T die Daten `Ex_data` L des arithmetischen Rechenergebnisses festhält, die in dem L-Zwischenspeicher 20 gespeichert sind. Ein Register der Register-Datei 9 wird gemäß einem in dem T-Zwischenspeicher 21 gespeicherten Befehl WB. IR ausgewählt und ein Datenwert WB_data, der in dem T-Zwischenspeicher 24 gespeichert ist, wird in das ausgewählte Register eingeschrieben. Das Zeichen "WB", vor den Befehlen, Daten und Kennungen, die in den Zwischenspeichern 21, 22, 23 und 24 gespeichert sind, zeigt an, daß sie sich in einer Rückschreibe-Einheit befinden.

Das parallel verarbeitende Gerät enthält weiterhin einen WB-Steuerkreis 100, der auf ein Anzeigesignal "taken" der Sprungbestimmung, auf ein "Nicht-Sprung"-Bestimmungs-Anzeigesignal "not taken" und auf ein eine Befehlsgültigkeit anzeigendes Signal "Ex can" hin, die von dem Befehls-Decodierer 10 erzeugt wurden den Zustand der WB-Einheit in einen der Zustände, Daten-Haltezustand, Daten-Schreibausführungszustand und Daten-Ungültigkeitszustand einstellt. Unter Bezugnahme auf Fig. 7 wird anschließend die Befehlsabgabe/Ausführungs-Abfolge des erfindungs-gemäßen parallel verarbeitenden Gerätes beschrieben.

Wenn der Befehlsdecoder 10 einen Befehl IR an eine entsprechende funktionelle Einheit abgibt, wird von ihm eine Schreibverzögerungs-Kennung `delay_inst` und eine Gültigkeits-Kennung `inst_avail` dem Befehl IR angehängt oder hinzugefügt. In jeder funktionellen Einheit werden diese Kennungen `delay_inst` und `inst_avail` über den Pipeline-Prozessor (die EX- und die WB-Einheit im Falle einer Ganzzahl-Einheit) zusammen mit dem Befehl IR übertragen und zur Steuerung des Umschreibens des Maschinenzustandes verwendet.

Die Gültigkeits-Kennung `inst_avail` ist eine Kennung, die anzeigt, ob ein vom Befehlsdecoder 10 abgegebener Befehl ein gültiger Befehl ist, der auszuführen ist. Beispielsweise ist in dem Fall, daß eine Befehlsabruf-Anforderung an den Befehlsspeicher erzeugt wird und kein Befehl von dem Befehlsspeicher zugeführt wird, kein gültiger Befehl dem Befehlsdecoder 10 zugeführt, so daß in diesem Zyklus kein gültiger Befehl zugeführt wird. Gemäß diesem Fall wird die Gültigkeits-Kennung `inst_avail` in den OFF-Zustand gebracht. Die Zustands-erkennung wird durch die Beobachtung des Bereit-

schafts-Signals IR_READY durchgeführt, daß von dem Befehlsspeicher mit der Befehlszuführung übergeben wird. Wenn die Gültigkeits-Kennung `inst_avail` im ON-Zustand ist, wird angezeigt, daß der Befehl gültig ist und daß er in einer funktionellen Einheit ausgeführt wird. Die Gültigkeits-Kennung `inst_avail` die dem Befehl IR hinzugefügt wird, der von dem Befehlsdecoder 10 an eine funktionelle Einheit abgegeben wird, wird auf "ON" gesetzt. Wie im folgenden beschrieben, werden die Kennungen `delay_inst` und `inst_avail` in jeder Einheit gesteuert.

1) Die Befehlsdecoder-Einheit (ID-Einheit):

a) eine Vielzahl von Befehlen, die der Befehls-Decodiereinheit ID zugeführt werden, sind in dem Befehlsdecoder 10 decodiert. In dem Fall, daß in den zugeführten Befehlen ein Sprungbefehl enthalten ist, und es nicht möglich ist zu entscheiden ob in dem Zyklus ein Sprung entsprechend dem Sprungbefehl erzeugt werden soll, überführt der Befehlsdecoder 10 die Schreibverzögerungs-Kennung `delay_inst` der Befehle nach dem Sprungbefehl, die an die funktionellen Einheiten abgegeben wurden, in den ON-Zustand und gibt sie zusammen mit der Gültigkeits-Kennung `inst_avail`, die in den ON-Zustand gebracht wurde und dem Befehlen IR ab, bis bestimmt ist, ob ein Sprung erzeugt wird.

b) In dem Fall, daß kein Befehl auf die Bestimmung wartet, ob ein Sprung mit den dem Befehlsdecoder zugeführten Befehlen erzeugt wird, ist es nicht notwendig, das Schreiben der Befehls-Ausführungsergebnisse zu verschieben. In diesem Fall wird die Schreibverzögerungs-Kennung `delay_inst` zu "OFF" gemacht und mit den Befehlen IR abgegeben.

c) In dem Fall, daß ein Sprungbefehl enthalten ist und bestimmt ist, daß ein Sprung demgemäß erzeugt wird, ist es nicht notwendig die Befehle nach dem Sprungbefehl auszuführen, so daß die Gültigkeits-Kennungen `inst_avail` zu "OFF" gesetzt werden. Dann werden die Befehle IR nicht an die funktionellen Einheiten abgegeben. Die Verzögerungs-Kennungen `delay_inst` werden ebenfalls in den OFF-Zustand gebracht.

Die Befehle IR und die Kennungen `delay_inst` und `inst_avail` werden über die EX-Einheiten der funktionellen Einheiten an die WB-Einheiten übermittelt. In den WB-Einheiten wird die folgende Steuerung durchgeführt.

2) Die WB-Einheit, in der das Neuschreiben (Wechseln) des Maschinen-Zustandes erfolgt:

a) In dem Fall, daß ein Befehl WB IR ein Befehl ist, der ein Neuschreiben des Maschinenzustandes durchführt und die Schreibverzögerungs-Kennung `delay_inst` "ON" ist, befindet sich der Befehl WB IR zum Bestimmen, ob ein Sprung erzeugt wird, in einem Wartezustand und eine Änderung des Maschinenzustandes hat nicht zu erfolgen. Daher wird das Neuschreiben des Maschinenzustandes offen gelassen und verzögert bis bestimmt ist, ob ein Sprung erzeugt ist.

b) In dem Fall, daß ein Sprung entsprechend eines Sprungbefehls erzeugt ist, werden ein Sprung anzeigendes Signal "taken" und ein Si-

gnal EX_{can} zum Ungültigmachen eines Befehls, der in der EX-Einheit steht, von dem Befehlsdecoder 10 erzeugt. Der WB-Steuerkreis 100 setzt die Gültigkeits-Kennung WB_{inst} avail eines Befehls WB IR, der die Kennungen WB_{delay} inst und WB_{inst} avail hat, die beide "ON" sind, in den OFF-Zustand. Dieses verursacht, daß der Befehl WB IR in der WB-Einheit ungültig wird und daß die Änderung des Maschinenzustandes durch den Befehl WB IR (das Datenschieben in die Register-Datei 9 in der in Fig. 7 dargestellten Struktur) verhindert ist. Dieses verursacht, daß nach dem Sprungbefehl an die funktionellen Einheiten abgegebene Befehle ungültig sind.

c) In dem Fall, in dem bestimmt ist, daß kein Sprung gemäß dem Sprungbefehl erzeugt ist, wird ein "Nicht-Sprung"-Erkennungs-Signal not_{taken} von dem Befehlsdecoder 10 erzeugt. Dann setzt der WB-Steuerkreis 100 die Schreibverzögerungs-Kennung WB_{delay} inst in den OFF-Zustand. Dies verursacht gemäß dem Befehl WB IR eine Änderung des Maschinenzustandes, der sich, für den Fall, daß ein Sprung erzeugt wird, zum Schutz gegen eine sofortige Ausführung in einem schwebenden Zustand befindet. Unter Bezugnahme auf Fig. 7 wird der Betrieb nun genau geschrieben.

Die von den Befehlsspeicher abgerufenen Befehle sind in dem Befehlsdecoder 10 decodiert. In dem Fall, daß das Decodierungsergebnis eine arithmetische Ganzzahl-Operation ist, werden von dem Befehlsdecoder 10 zwei Register-Bezeichnungssignale (Register-Zeiger) zu der Register-Datei 9 erzeugt, und aus dieser werden zwei Daten, data 1 und data 2, gelesen, die Daten für eine arithmetische Operation sind.

Der Befehlsdecoder 10 gibt als Decodierungsergebnis einen Befehl IR, eine Gültigkeits-Kennung inst_{avail}, daß im ON-Zustand anzeigt, daß der Befehl IR gültig ist, und eine Schreibverzögerungs-Kennung delay_{inst} ab. Die Schreibverzögerungs-Kennung delay_{inst} ist gemäß dessen gesetzt, ob der Befehl IR ein Befehl nach einem Sprungbefehl ist und ob bestimmt werden kann, ob ein Sprung gemäß dem Sprungbefehl in diesem Zyklus erzeugt wird. Im Fall eines Wartezustandes zur Bestimmung ob ein Sprung gemäß dem Sprungbefehl erzeugt wird, wird die Schreibverzögerungs-Kennung delay_{inst} in den ON-Zustand (binär "1") gebracht. In dem Fall, daß kein Sprungbefehl enthalten ist oder in dem Fall, daß bestimmt ist, daß ein Sprung gemäß dem Sprungbefehl erzeugt wird, ist die Schreibverzögerungs-Kennung delay_{inst} in den OFF-Zustand (binär "0") gesetzt. In dem Fall, daß bestimmt ist, daß ein Sprung gemäß einem Sprungbefehl in diesem Zyklus erzeugt ist, und der Befehl IR nicht notwendig ist, wird die Gültigkeits-Kennung inst_{avail} in den OFF-Zustand gebracht.

Eine Ausgabe des Befehlsdecoders 10 und eine Ausgabe der Register-Datei 9 werden als Antwort auf das Zeittaktsignal T in den T-Zwischenspeichern 11 bis 15 gehalten. Die in den T-Zwischenspeichern 14 und 15 gehaltenen Daten tmp1 und tmp2 (entsprechend den Daten data 1 und data 2) werden dem Rechen- und Steuerwerk (ALU) 16 zugeführt und eine vorbestimmte vorgeschriebene arithmetische Operation wird durchgeführt.

Die in den T-Zwischenspeichern 11 bis 13 gehaltenen

Befehle und Kennungen werden als Antwort auf das Zeittaktsignal L in den entsprechenden L-Zwischenspeichern 17 bis 19 gehalten. Eine Ausgabe des Rechen- und Steuerwerkes (ALU) 16 wird als Antwort auf das Zeittaktsignal L in dem L-Zwischenspeicher 20 gehalten. Dies führt zu einer Vervollständigung der Operation in der EX-Einheit.

In dem Fall, in dem kein Befehl in der WB-Einheit steht, werden als Antwort auf das Zeittaktsignal T die Befehle, Kennungen, und die in den L-Zwischenspeichern 17-20 gehaltenen Daten in den T-Zwischenspeichern 21-24 festgehalten. In dem Fall, in dem ein Befehl (d. h. ein Befehl der auf die Bestimmung wartet, ob ein Sprung erzeugt wird) in der WB-Einheit steht, hält die EX-Einheit, gesteuert von dem WB-Steuerkreis 100, weiterhin die Befehle, die Kennungen, und die Daten fest.

In dem Fall, daß die in dem L-Zwischenspeicher 23 gehaltene Kennung WB_{delay} inst in der WB-Einheit im ON-Zustand ist, wird das Datenschieben in die Register-Datei 9 durch die Kontrolle des WB-Steuerkreises 100 in einem Zustand gebracht, der die Änderung des Maschinenzustandes verzögert, und die zugeführten Daten bleiben darin festgehalten.

Falls bestimmt ist, daß in dem Befehlsdecoder 10 ein Sprung gemäß dem Sprungbefehl erzeugt ist, wird ein eine Sprungbestimmung anzeigendes Signal taken erzeugt und dem WB-Steuerkreis 100 zugeführt. Der WB-Steuerkreis 100 bringt als Antwort auf das Signal taken die Schreibverzögerungs-Kennung WB_{delay} inst in den OFF-Zustand und die Gültigkeits-Kennung WB_{inst} avail in den OFF-Zustand. Dies verursacht, daß der in dem T-Zwischenspeicher 21 gehaltene Befehl WB IR ungültig wird und daß die in dem T-Zwischenspeicher 24 gehaltenen Daten WB data ungültig werden, und daß das Schreiben in die Register-Datei 9 nicht durchgeführt wird.

Falls andererseits bestimmt ist, daß gemäß dem Sprungbefehl kein Sprung in dem Befehlsdecoder erzeugt ist, wird ein eine "Nicht-Sprungbestimmung" erkennendes Signal not_{taken} erzeugt und dem WB-Steuerkreis 100 zugeführt.

Der WB-Steuerkreis 100 bringt die in dem T-Zwischenspeicher festgehaltene Schreibverzögerungs-Kennung WB_{delay} inst als Antwort auf das Signal not_{taken} in den OFF-Zustand. Dann wird gemäß dem in dem T-Zwischenspeicher 21 gehaltenen Befehl WB IR die Wahl eines Registers der Register-Datei 9 durchgeführt und die Daten WB data werden in das ausgewählte Register geschrieben.

Wenn ein eine Sprungbestimmung bestimmendes Signal taken in dem Befehlsdecoder 10 erzeugt ist wird gleichzeitig ein Signal EX_{can} zum Ungültigmachen des in der EX-Einheit festgehaltenen Befehls EX IR L erzeugt. Die in dem L-Zwischenspeicher 18 gehaltene Gültigkeits-Kennung EX_{inst} avail L wird durch die Steuerung des WB-Steuerkreises 100 als Antwort auf das Signal EX_{can} in den OFF-Zustand gebracht, und das Ungültigmachen des in der EX-Einheit gehaltenen Befehls durchgeführt. Es folgt die Beschreibung einer Struktur, mit der die Vorbereitung eines Sprungverfahrens und das Ungültigmachen eines Befehls bei einer zuvor beschriebenen Sprungbestimmung durchzuführen ist.

Fig. 9 zeigt ein Diagramm in dem ein Beispiel einer logischen Struktur in einer Maschinenzustands-Änderungs-Einheit (WB-Einheit) dargestellt ist. Unter Bezugnahme auf Fig. 9 enthält die WB-Einheit einen Logik-

Schaltkreis 30, der in der dem T-Zwischenspeichers 21 vorangehenden Einheit enthalten ist, ein Logik-Schaltkreis 31, der in der vorangehenden Einheit dem T-Zwischenspeicher 22 enthalten ist, ein Logik-Schaltkreis 32 der in der dem T-Zwischenspeicher 23 vorangehenden Einheit enthalten ist, und ein Logik-Schaltkreis 33, der in der dem T-Zwischenspeicher 24 vorangehenden Einheit enthalten ist.

Als Antwort auf ein Signal WB_busy, das anzeigt, daß die WB-Einheit sich in einem Maschinenzustands-Änderungsverzögerungs-Zustand befindet, d. h., daß Sprungbestimmungs-Wartezustand, daß der in der WB-Einheit gehaltene Befehl ungültig ist, passiert selektiv entweder ein Befehl EX_IR_L oder ein Befehl WB_IR_L den Logikkreis 30.

Als Antwort auf das Signal WB_busy das Signal WB_can und ein Signal EX_can, das anzeigt, daß der Befehl in der EX-Einheit ungültig ist, läßt der Logik-Schaltkreis 31 selektiv entweder eine Gültigkeits-Kennung EX_inst_avail_L aus der EX-Einheit oder eine Gültigkeits-Kennung WB_inst_avail_L der WB-Einheit passieren.

Als Antwort auf das Signal WB_busy und WB_can läßt der Logik-Schaltkreis 32 entweder eine Schreibverzögerungs-Kennung EX_delay_inst_L aus der EX-Einheit oder eine Schreibverzögerungs-Kennung WB_delay_inst_L der WB-Einheit passieren.

Als Antwort auf das Signal WB_busy und WB_can läßt der Logik-Schaltkreis 33 selektiv entweder einen Datenwert EX_data_L aus der EX-Einheit oder einen Datenwert WB_data_L der WB-Einheit passieren.

Die WB-Einheit enthält weiterhin einen Befehlsdecoder 34 zum Decodieren des in dem T-Zwischenspeicher 21 festgehaltenen Befehls WB_IR und zum Erzeugen eines Schreibanzeige-Signals w_RF, das anzeigt, ob die Daten in die Register-Datei 9 zu schreiben sind, und einen Register-Zeiger address der Register-Datei 9 und einen Logik-Schaltkreis 29 zum Überwachen des Zustandes der WB-Einheit, um die WB-Einheit in einen der Zustände, den Datenhaltezustand, den Daten-Ungültigkeitszustand und den Daten-Schreibausführungs-Zustand zu setzen.

Der Logik-Schaltkreis 29 erzeugt die Signale WB_busy und WB_can und ein Signal wen, um als Antwort auf ein Registerdatei-Schreibanzeigesignal w_RF des Befehlsdecoder 34, auf ein Punktbestimmungssignal taken vom Befehlsdecoder 10, auf ein "Nicht-Sprung"-Bestimmungssignal not_taken, auf eine Gültigkeits-Kennung WB_inst_avail_L und auf eine Schreibverzögerungs-Kennung WB_delay_inst_L, die Register-Datei 9 in einen Schreibaktivierungs-Zustand zu setzen.

Der Logik-Schaltkreis 31 führt eine logische Operation aus, die durch "in_logic 1" angezeigt ist. Die Logik-Schaltkreise 30, 32 und 33 führen eine logische Operation durch, die durch "in_logic 2" angezeigt ist. Der Logik-Schaltkreis 29 führt eine logische Operation aus, die durch WB_busy_logic angezeigt ist. Jetzt werden die durch die entsprechenden Logik-Schaltkreise eingeführten logischen Strukturen beschrieben.

Fig. 10 zeigt tabellarisch in einem Diagramm die Logik der WB_busy_logic, die mit dem Logik-Schaltkreis 29 eingeführt ist. Die Logik des Logik-Schaltkreises 29 wird im folgenden beschrieben.

1) Der Fall, daß sich kein gültiger Befehl in der WB-Einheit befindet:

Bei dieser Bedingung, hält die WB-Einheit keine zu schreibenden Daten fest, so daß sie keine Opera-

tion ausführt. Speziell in dem Fall, daß WB_inst_avail_L "off" (0) ist, sind alle Signale "wen", "WB_busy" und "WB_can" "off" (0).

2) Der Fall, daß in der WB-Einheit ein gültiger Befehl festgehalten wird:

Bei dieser Bedingung befindet sich "WB_inst_avail_L" in einem ON-Zustand (1).

a) In dem Fall, daß das Signal w_RF aus dem Befehlsdecoder 34 im OFF-Zustand ist, wird das Datenschieben in die Register-Datei nicht durchgeführt. Solch ein Zustand ergibt sich beispielsweise für den Fall, daß die Bedingung für eine einfache Befehlsausführung bestimmt wird. Insbesondere wird ein solcher Zustand dann erreicht, wenn auf der Grundlage irgendeines Vergleichsergebnisses ein Verfahren bestimmt ist, da es beispielsweise nicht notwendig ist das Vergleichsergebnis in das Register zu schreiben.

In diesem Fall wird das Datenschieben nicht durchgeführt und es ist möglich einem neuen Befehl in dem nächsten Zyklus abzurufen. Demgemäß sind in dem Fall, in dem "w_RF" "OFF" ist, alle Signale "wen", WB_busy, und WB_can in den OFF-Zustand gesetzt.

b) Der Fall, daß das Signal w_RF "ON" ist und das Kennung WB_delay_inst_L "OFF" ist:

In diesem Fall besteht in der WB-Einheit ein gültiger Befehl und diese befindet sich nicht in einem auf eine Sprungbestimmung wartenden Zustand, so daß das Datenschieben ausgeführt werden darf. Demgemäß wird in dem Fall, daß "w_RF" "ON" ist und "WB_delay_inst_L" "OFF" ist, das Signal wen zu "ON", und die Signale WB_busy und WB_can werden zu "OFF".

c) Der Fall, daß das Signal w_RF und die Kennung WB_delay_inst_L "ON" sind:

In diesem Fall ist angezeigt, daß ein Befehl, der ein Datenschieben ausführen soll, sich in der WB-Einheit befindet und daß der Befehl ein Befehl nach einem Sprungbefehl ist.

1) In dem Fall, in dem nicht entschieden ist, ob in diesem Zustand ein Sprung in dem Befehlsdecoder erzeugt wird, wird die WB-Einheit in einen Zustand gebracht, der die Maschinen-Zustandsänderung verschiebt. Demgemäß werden in dem Fall, daß sich beide die Signale not_taken und taken im OFF-Zustand befinden, die Signale wen und WB_can zu "OFF", und das Signal WB_busy zu "ON".

2) Der Fall, daß bestimmt ist, daß in einem auf eine Sprungdurchführung wartenden Zustand kein Sprung erzeugt wird:

In diesem Fall kann der Befehl in der WB-Einheit sofort ausgeführt werden. Demgemäß werden für den Fall, daß das Signal not_taken "ON" ist und das Signal taken "OFF" ist, das Signal wen zu "ON", und WB_busy und WB_can beide zu "OFF".

3) Der Fall, daß bestimmt ist, das in dem Warte-Zustand ein Sprung erzeugt ist:

In diesem Fall ist der in der WB-Einheit festgehaltene Befehl ungültig. Das Datenschieben in die Register-Datei wird nicht ausgeführt und in einen Zustand gebracht, um auf einen in einem späteren Zyklus abgegebenen gültigen Befehl zu warten.

Demgemäß werden für den Fall, daß das Signal

not taken "OFF" ist und das Signal taken "ON" ist, die Signale wen und WB_busy zu "OFF" und das Signal WB_can wird zu "ON".

Die spezielle Schaltkreisstruktur, die die Logik der in Fig. 10 gezeigten "WB_busy_logic" ermöglicht ist nicht dargestellt, da sie basierend auf der in Fig. 10 gezeigten Tabelle auf einfache Weise ermöglicht werden kann. Die Logik kann durch die Verwendung von Gatterschaltkreisen, sowohl von UND-Gatter und ODER-Gatter als auch von PLD-Elementen (programmierbare logische Elemente) ermöglicht werden.

Fig. 11 zeigt in einer Tabelle das Diagramm einer Logik der "in logic 1". Die "in logic 1" wird durch den logischen Schaltkreis 31 ausgeführt. Die Logik der "in logic 1" wird unter Bezugnahme auf Fig. 11 im folgenden beschrieben.

1) Der Fall, daß "WB_can" sich im ON-Zustand befindet:

Dieser Zustand zeigt an, in welchem Zustand der Wartebefehl zur Sprungbestimmung in der WB-Einheit ungültig gemacht ist. Ob die WB-Einheit im nächsten Zyklus einen gültigen Befehl festhält hängt davon ab, ob die EX-Einheit in Zyklus ungültig ist. Wenn das Signal EX_can "OFF" ist, wird der vorliegende Befehl EX-Einheit in dem nächsten Zyklus der WB-Einheit zugeführt. Die Gültigkeit des Befehles wird gemäß der Kennung EX_inst_avail L bestimmt. Demgemäß wird in dem Fall, daß das EX_can sich im OFF-Zustand befindet, die Kennung WB_avail gleich "EX_inst_avail L".

In dem Fall, daß die Kennung EX_can "ON" ist, ist angezeigt, daß der Befehl in der EX-Einheit ungültig ist. Demgemäß wird im nächsten Zyklus von der EX-Einheit kein Befehl in die WB-Einheit zugeführt. Insbesondere in dem Fall, daß EX_can "ON" ist, wird "WB_inst_avail" "OFF".

2) Der Fall, in dem "WB_can" "OFF" ist: Dieser Zustand zeigt an, daß die Befehls-Ungültigkeit-Operation in der WB-Einheit nicht ausgeführt wurde.

In Fig. 12 ist in einem Diagramm die mittels der "in logic 2" eingeführte Logik gezeigt. Die Logik der "in logic 2" wird unter Bezugnahme auf Fig. 12 nachfolgend beschrieben.

In dem Fall, daß das Signal WB_busy im OFF-Zustand ist wird angezeigt, daß kein Befehl vorliegt, der in dem nächsten Zyklus in der WB-Einheit festzuhalten ist, und daß der Befehl aus der EX-Einheit in die WB-Einheit genommen werden kann. Demgemäß wird, wenn das Signal WB_busy "OFF" ist, die Kennung WB_inst_avail gleich der Kennung EX_inst_avail L.

In dem Fall, daß das Signal WB_busy sich im ON-Zustand befindet, wird angezeigt, daß die WB-Einheit einen Befehl im Wartezustand festhält, um zu bestimmen, ob ein Sprung erzeugt wird. Demgemäß ist es notwendig die Gültigkeits-Kennung dieses Zyklus ebenfalls im nächsten Zyklus festzuhalten. Insbesondere wird die Kennung WB_inst_avail gleich der Kennung WB_inst_avail L.

Das Vorsehen des Logik-Schaltkreises, der die zuvor beschriebene logische Operation in der WB-Einheit ausführt ermöglicht es, Daten für den Fall mit hoher Geschwindigkeit in das Daten-Register zu schreiben, daß bestimmt ist, daß im Wartezustand auf die Bestimmung, ob ein Sprung gemäß dem Sprungbefehl erzeugt wird,

kein Sprung erzeugt wird. Diese Operation wird im folgenden unter Bezugnahme auf ein Operationskurven-Diagrammes beschrieben.

Fig. 13 stellt in einem Diagramm eine Signal-Kurvenform der Operation in der WB-Einheit für den Fall dar, daß im Wartezustand auf die Bestimmung, ob ein Sprung erzeugt wird bestimmt ist, daß ein Sprung erzeugt wird. Die Operation für den Fall, daß bestimmt ist, daß ein Sprung erzeugt ist, wird im folgenden unter Bezugnahme auf Fig. 13 beschrieben.

Zyklus 0:

In diesem Zyklus wird der WB-Einheit ein Befehl zugeführt, der ein Befehl nach einem Sprungbefehl ist, und der auf die Bestimmung des Zustandes des Sprungbefehles warten soll. Die Schreibverzögerungs-Kennung WB_delay_inst des Befehles ist im ON-Zustand. Die Gültigkeits-Kennung WB_inst_avail des Befehles ist auch im ON-Zustand. Die Zustände der Ausgaben WB_delay_inst L und WB_inst_avail L der Logikschaltkreise 26 und 27 sind als Antwort auf den Zeittakt L festgesetzt. Darauf als Antwort ist das Signal WB_busy des Logikschaltkreises 29 in den ON-Zustand gebracht.

Hierbei wird ein Zustand angenommen, bei dem ein der WB-Einheit zugeführter Befehl ein Befehl zum Datenschieben in die Register-Datei 9 ist und wobei als Antwort auf das Zeittaktsignal T vom Befehlsdecodierer 34 ein Signal w_RF erzeugt wird. Der logische Schaltkreis 29 erzeugt gemäß dem Wartezustand für die Sprungbestimmung kein Schreib-Aktivierungs-Signal wen (zu "0" gesetzt) und setzt es in den Datenschreibunterbindungs-Zustand.

Normalerweise hat die Registerdatei 9 die Aufgabe als Antwort auf das Signal wen bei dem Zeittaktsignal L die Daten WB_data L des Zwischenspeichers zu schreiben. Der Registerzeiger address wird jedoch als Antwort auf das Zeittaktsignal T festgesetzt. Das Datenschieben in die Register-Datei 9 befindet sich in dem Zyklus 0 im unterbundenen Zustand.

Andererseits ändern die Logikschaltkreise 30, 31, 32 und 33 ihre Eingabepfade als Antwort auf das ON-Zustands-Signal WB_busy und wählen die Ausgänge der entsprechenden L-Zwischenspeicher 25, 26, 27 und 28, um sie den entsprechenden T-Zwischenspeichern 21, 22, 23 und 24 entsprechend zu übermitteln.

Der Zyklus 1:

In diesem Zyklus befindet sich das Signal WB_busy im ON-Zustand, so daß der Befehl, die Kennung und die der WB-Einheit im Zyklus 0 zugeführten Daten, in den T-Zwischenspeichern 21 bis 24 festgehalten werden.

Der Zyklus 2:

In diesem Zyklus wird festgestellt, daß ein Sprung erzeugt ist und es wird festgestellt, daß der Befehl in der WB-Einheit ungültig zu machen ist. In diesem Fall wird das Signal taken in den ON-Zustand gebracht und dem logischen Schaltkreis 29 zugeführt. Dieses verursacht die Unterbindung der Erzeugung eines Schreib-Aktivierungs-Signals wen und die Unterbindung des Datenschiebens in die Register-Datei. Das Signal WB_busy ist in den OFF-Zustand gesetzt und das Signal WB_can ist als Antwort auf das Signal taken in den ON-Zustand gesetzt. Die WB-Einheit wird in einen Zustand gebracht,

in dem der Befehl von der EX-Einheit als Antwort auf den OFF-Zustand des Signals WB_busy zu empfangen ist.

In dem Fall, daß der in der EX-Einheit festgehaltene Befehl ein Befehl nach dem Sprungbefehl ist, der sich zu diesem Zeitpunkt nicht auf den Sprungziel-Befehl bezieht, wird ein Signal EX_can zum Ungültigmachen des in der EX-Einheit festgehaltenen Befehls erzeugt. Demgemäß entsprechen die Kennungen WB_inst_avail und WB_delay_inst des Zyklus 3 den Kennungen des im Zyklus 2 in der EX-Einheit gehaltenen Befehls.

In dem Zyklus 3 sind alle Signale WB_busy, taken und WB_can in den OFF-Zustand gebracht und die WB-Einheit führt den von der EX-Einheit empfangenen Befehl aus.

Die "in logic 2"-Logik der Logikschaltkreise 30, 32 und 33 empfangen ebenfalls das Signal WB_can an ihrem Eingang, während die Logik nur gemäß dem Signal WB_busy der zuvor beschriebenen logischen Operation gesetzt ist. Es kann eine Struktur eingesetzt werden, in der das Signal WB_can ebenfalls für die logische Operation verwendet wird. In dieser Struktur werden der Befehl, die Kennung und die Daten in der WB-Einheit festgehalten, wenn das Signal WB_busy "ON" ist und das Signal WB_can im OFF-Zustand ist, während der Befehl, die Daten und die Kennung aus der EX-Einheit passieren, ohne den ON/OFF-Zustand des Signals WB_can zu beachten, wenn das Signal WB_busy "OFF" ist. Eine Schaltungsausführung für den Fall, daß beide der zwei Signale WB_busy und WB_can verwendet werden, ist in der Zeichnung gezeigt.

Obwohl das Signal EX_can immer in dem Fall erzeugt wird, wenn das Signal taken erzeugt ist, gibt es einen möglichen Fall, in dem es notwendig ist, die in der EX-Einheit festgehaltenen Daten ungültig zu machen, auch wenn kein Sprung erzeugt ist. In solch einem Fall ist das Signal EX_can erzeugt und es erfolgt das Ungültigmachen des Befehls in der EX-Einheit. Dies ist auch für das Signal WB_can der Fall.

Bezugnehmend auf die in Fig. 13 dargestellte Signal-Kurvenform weist die Zeitgebung des Anstiegs und Abfallens des Signals wen unterschiedliche Weiten auf und die Zeitgebung des Fallens des Signals taken hat ebenfalls eine verschiedene Weite. Dieses zeigt an, daß jedes Signal innerhalb dieser Zeitbreite mit einer geeigneten Zeitgebung ansteigen oder fallen kann. Insbesondere das Signal wen kann mit der Zeitgebung aktiviert werden, die das Schreiben der Daten WB_data_L in den festgesetzten Zustand aktiviert und der inaktive Zustand des Signals wen kann mit einer Zeitgebung festgesetzt werden, mit der er vor dem Daten-Schreib-Bestimmungszustand in den festgesetzten Zustand kommt.

Während das Signal taken mit der Zeitgebung des Zeittaktsignals L erzeugt wird, kann die Zeitgebung des Falles zum Definieren der Ausführungsoperation des Abrufverfahrens eines Sprungziel-Befehls des nächsten Zyklus gesetzt werden, wenn bestimmt ist, daß ein Sprung erzeugt wird, und es kann beispielsweise wie zum Beginn des Zyklus 3 in Fig. 13 in dem festgesetzten Zustand gehalten werden. Unter Bezugnahme auf Fig. 14 wird nun die Bestimmungs-Operation eines "Nicht-Sprunges" beschrieben.

Fig. 14 erläutert in einem Kurvenform-Diagramm die Operation in dem Fall, in dem bestimmt ist, daß kein Sprung in dem Wartezustand zur Sprung-Zustands-Bestimmung in der WB-Einheit erzeugt wird.

Der Zyklus 0 und der Zyklus 1 sind die gleichen wie die in Fig. 13 dargestellten Zyklus 0 und Zyklus 1. Falls

bestimmt ist, daß kein Sprung in dem Zyklus 2 erzeugt ist, wird das Signal not_taken in den ON-Zustand gebracht. Als Antwort darauf wird das Schreib-Aktivierungs-Signal wen in den ON-Zustand gebracht und das Datenschreiben in die Register-Datei 9 aktiviert. Die Register-Datei 9 schreibt die Daten WB_data_L des L-Zwischenspeichers 28 in ein Register, daß durch den Registerzeiger address vom Befehls-Decodierer 34 als Antwort auf das Signal wen ausgewählt ist.

Zu diesem Zeitpunkt ist der Befehl in der WB-Einheit ausgeführt und die Änderung des Maschinen-Zustands durchgeführt, so daß die WB-Einheit in den Zustand gebracht ist, in dem der nächste Befehl von der EX-Einheit empfangen werden kann. Demgemäß ist das Signal WB_busy in den OFF-Zustand gebracht. Das Signal WB_can bleibt im OFF-Zustand. Dies bringt jeden der Logikschaltkreise 30 bis 33 in einen Zustand, in dem die Ausgabe des entsprechenden L-Zwischenspeichers die EX-Einheit passiert.

In der Operation des 3. Zyklus wird die Ausführung entsprechend dem Befehl aus der EX-Einheit durchgeführt. Insbesondere, falls der von der EX-Einheit zugeführte Befehl ein ungültiger Befehl ist, wird keine Ausführung durchgeführt und falls er ein gültiger Befehl ist, wird die befehlsgemäße Durchführung ausgeführt. Die Logik und die Operation der EX-Einheit wird im folgenden beschrieben.

Fig. 15 zeigt in einem Blockschaltbild die Struktur einer EX-Einheit. In Fig. 15 sind die Teile, die den Teil der in Fig. 7 dargestellten Struktur entsprechen mit den selben Bezugszeichen ausgeführt. Unter Bezugnahme auf Fig. 15 enthält die EX-Einheit logische Schaltkreise 41, 42, 43, 44 und 45, die in den vorangehenden Einheiten der T-Zwischenspeichern 11, 12, 13, 14 und 15 vorgesehen sind und halten die L-Zwischenspeicher 35 und 36 die in den T-Zwischenspeichern 14 und 15 festgehaltenen Daten entsprechend als Antwort auf das Zeittaktsignal L fest.

Als Antwort auf das Signal WB_busy, wie in Fig. 9 dargestellt, passiert den logischen Schaltkreis 41 entweder der Befehl IR aus dem Befehlsdecoder 10 der Decodier-Einheit 1D oder der Befehl EX_IR_L des L-Zwischenspeichers 17.

Als Antwort auf das Signal WB_busy passiert den logischen Schaltkreis 42 entweder die Gültigkeits-Kennung inst_avail aus dem Befehlsdecoder 10 oder die Gültigkeits-Kennung EX_inst_avail_L aus dem L-Zwischenspeicherkreis 18'.

Den logischen Schaltkreis 43 passiert entweder die Schreibverzögerungs-Kennung delay_inst aus dem Befehlsdecoder 10 oder die Kennung EX_delay_inst_L aus dem L-Zwischenspeicher 19.

Die Logikschaltkreise 44 und 45 werden entweder von den Daten data1 und data2, die aus der Register-Datei gelesen wurden oder entsprechend von den in den L-Zwischenspeichern 35 und 36 gehaltenen Daten tmp1_L und tmp2_L passiert.

Der L-Zwischenspeicher 18' hat als Antwort auf das Signal EX_can seine festgehaltenen Daten in den OFF-Zustand rückgesetzt und gesetzt. Die Steuerung der festgehaltenen Daten des L-Zwischenspeichers 18' kann gemäß dem Signal EX_can durch eine Struktur ermöglicht werden, bei der in dem Eingang des L-Zwischenspeichers 18 ein Schaltkreis vorgesehen ist, mit dem der Ausgangswert des Schaltkreises als Antwort auf das Signal EX_can zu "EX_inst_avail" oder "0" gesetzt werden kann, falls das Signal EX_can mit der Erzeugung des Zeittaktsignals L in den festgesetzten Zustand gebracht

ist.

Die Logikschaltkreise 41-45 führen dieselbe Logik ein und führen die mit "in_logic 3" bezeichnete Logik aus.

In Fig. 16 ist tabellarisch ein Diagramm gezeigt, daß die Logik der "in_logic 3" in einem Diagramm zeigt. Die Logik der "in_logic 3" wird im folgenden unter Bezugnahme auf Fig. 15 und 16 beschrieben.

In dem Fall, daß die WB-Einheit sich in dem Befehls-Haltezustand befindet, führt der Zustand dazu, daß die EX-Einheit keinen Befehl der WB-Einheit zuführen kann und in den Befehls-Haltezustand gebracht wird. Dieser Zustand ist durch das Signal WB_busy im ON-Zustand (binär 1) angezeigt. Demgemäß passieren in dem Fall, in dem das Signal WB_busy "ON" (binär 1) ist, die Ausgangssignale der L-Zwischenspeicher 17, 18', 19, 35 und 36 die entsprechenden logischen Schaltkreise 41 bis 45 zu den entsprechenden T-Zwischenspeichern 11 bis 15.

Andererseits in dem Fall, in dem das Signal WB_busy "OFF" (binär 0) ist, kann die WB-Einheit die gehaltenen Befehle der WB-Einheit zuführen und empfängt von dem Befehlsdecoder 10 einen neuen Befehl. Demgemäß für den Fall, daß das Signal WB_busy "OFF" ist, passieren die Ausgabe des Befehlsdecoder 10 und die Ausgabe der Register-Datei 9 die logischen Schaltkreise 41 bis 45.

Fig. 17 zeigt tabellarisch in einem Diagramm den Operations-Zustand des L-Zwischenspeichers 18'. Die Operation des L-Zwischenspeichers 18' wird im folgenden unter Bezugnahme auf Fig. 17 beschrieben. Falls das Signal EX_can in den ON-Zustand (binär 1) gesetzt ist, ist angezeigt, daß der in der EX-Einheit gehaltene Befehl ungültig ist. Demgemäß, falls das Signal EX_can "ON" ist, dreht die Ausgangs-Kennung EX_inst_avail_L des L-Zwischenspeichers 18' zu "OFF" (binär 0).

Andererseits in dem Fall, daß das Signal EX_can im OFF-Zustand (binär 0) ist, ist angezeigt, daß es nicht notwendig ist den in der EX-Einheit gehaltenen Befehl ungültig zu machen. In diesem Fall hält der L-Zwischenspeicher 18' die Gültigkeits-Kennung EX_inst_avail des T-Zwischenspeichers 12. Insbesondere in dem Fall, in dem das Signal EX_can "OFF" ist, ist die Kennung EX_inst_avail_L die Gültigkeits-Kennung EX_inst_avail des Zyklus.

Gemäß der zuvor beschriebenen Struktur ist es möglich gemäß dem Befehlsausführungszustand der WB-Einheit ein Befehl nach dem Sprungbefehl in der EX-Einheit auszuführen und zu halten. Das Signal WB_busy kann ebenfalls dem Befehlsdecoder 10 zugeführt werden, um ihn zur Steuerung der Einstellung (Verzögerung) der Befehlsabgabe zu benutzen.

Während eine Struktur zum Durchführen des Halten/Ungültigmachens des Befehls einer Ganzzahl-Einheit im vorangegangenen Ausführungsbeispiel beschrieben wurde, ist es möglich den selben Effekt in anderen funktionellen Einheiten wie einer Fließkomma-Arithmetik-Bearbeitungseinheit und einer Speicherzugriffs-Einheit durch die Anwendung derselben Struktur auf ihre Maschinenzustands-Änderungszustände zu erhalten. Zusätzlich ist die Anzahl der funktionellen Einheit nicht auf 4 begrenzt.

Erfindungsgemäß erfolgt eine Strukturierung wie zuvor beschrieben in der Art, daß für einen Befehl nach einem Sprungbefehl eine Schreibverzögerungs-Kennung angezeigt, daß er ein Befehl nach dem Sprungbefehl ist, der zusammen mit dem Befehl an eine funktionelle Einheit abgegeben wurde, und daß die Maschinenzu-

standsänderungs-Operation in der funktionellen Einheit gemäß der Schreibverzögerungs-Kennung gesteuert wird. Demgemäß ist auch, falls nicht bestimmt ist ob ein Sprung gemäß dem Sprungbefehl erzeugt ist, es möglich, den Befehl nach dem Sprungbefehl an die funktionelle Einheit abzugeben und wenn bestimmt ist, daß kein Sprung erzeugt ist, ist es möglich die Maschinenzustandsänderung sofort durchzuführen, so daß es möglich ist, ein parallel verarbeitendes Gerät mit wesentlich verbesserter Verarbeitungsgeschwindigkeit vorzusehen. Obwohl die Erfindung einzelnen beschrieben und dargestellt wurde ist klar verständlich, daß dieses nur der Darstellung und als Beispiel ohne Beschränkung dient und der Umfang und die Sichtweise der Erfindung nur mit dem Wortlaut der beiliegenden Ansprüche beschränkt ist.

Ein Prozessor vom Superskalaren-Typ enthält einen Befehlsspeicher, eine Abruf-Einrichtung, die gleichzeitig eine Vielzahl von Befehlen aus dem Befehlsspeicher abruf, funktionelle Einheiten die entsprechende vorbestimmte Funktion ausführen, und eine Decodier-Einrichtung, die die abgerufenen Befehle decodiert und die Befehle an die entsprechende funktionellen Einheiten abgibt, die parallel verarbeitet werden können. Die Decodiereinrichtung enthält einen Decodierer, der bestimmt, ob ein Sprung in den empfangenen Befehlen enthalten ist und ob ein Sprung gemäß dem Sprungbefehl erzeugt wird. Der Decodierer hängt mit der Befehlsabgabe an die funktionelle Einheit an den Befehl eine Schreibverzögerungs-Kennung, die anzeigt, ob der Befehl nach einem Sprungbefehl kommt und eine Gültigkeits-Kennung an, die anzeigt, ob der Befehl gültig ist. Die funktionelle Einheit enthält eine Ausführungseinheit, die einen Befehl ausführt und eine Rückschreib-Einheit die den Maschinenzustand gemäß dem Ausführungsergebnis in der Ausführungseinheit ändert. Ein Prozessor vom Superskalaren-Typ enthält einen Steuerschaltkreis der das Ändern des Maschinenzustandes durch die Rückschreib-Einheit unterbindet, wenn ein Sprung gemäß dem Sprungbefehl erzeugt wird. Die Steuerungseinrichtung setzt die Rückschreib-Einrichtung in einen Zustand der die Änderung des Maschinenzustandes verzögert, wenn noch nicht bestimmt ist, ob ein Sprung gemäß dem Sprungbefehl erzeugt ist und führt die Änderung des Maschinenzustandes mit der Rückschreib-Einheit aus, wenn bestimmt ist, daß kein Sprung gemäß dem Sprungbefehl erzeugt wird.

Ein Prozessor vom Superskalaren-Typ enthält einen Befehlsspeicher, eine Abruf-Einrichtung, die gleichzeitig eine Vielzahl von Befehlen aus dem Befehlsspeicher abruf, funktionelle Einheiten die entsprechende vorbestimmte Funktion ausführen, und eine Decodier-Einrichtung, die die abgerufenen Befehle decodiert und die Befehle an die entsprechende funktionellen Einheiten abgibt, die parallel verarbeitet werden können. Die Decodiereinrichtung enthält einen Decodierer, der bestimmt, ob ein Sprung in den empfangenen Befehlen enthalten ist und ob ein Sprung gemäß dem Sprungbefehl erzeugt wird. Der Decodierer hängt mit der Befehlsabgabe an die funktionelle Einheit an den Befehl eine Schreibverzögerungs-Kennung, die anzeigt, ob der Befehl nach einem Sprungbefehl kommt und eine Gültigkeits-Kennung an, die anzeigt ob der Befehl gültig ist. Die funktionelle Einheit enthält eine Ausführungseinheit, die einen Befehl ausführt und eine Rückschreib-Einheit die den Maschinenzustand gemäß dem Ausführungsergebnis in der Ausführungseinheit ändert. Ein Prozessor vom Superskalaren-Typ enthält einen Steu-

erschaltkreis der das Ändern des Maschinenzustandes durch die Rückschreib-Einheit unterbindet, wenn ein Sprung gemäß dem Sprungbefehl erzeugt wird. Die Steuerungs-Einrichtung setzt die Rückschreib-Einrichtung in einen Zustand der die Änderung des Maschinenzustandes verzögert, wenn noch nicht bestimmt ist, ob ein Sprung gemäß dem Sprungbefehl erzeugt ist und führt die Änderung des Maschinenzustandes mit der Rückschreib-Einheit aus, wenn bestimmt ist, daß kein Sprung gemäß dem Sprungbefehl erzeugt wird.

Patentansprüche

1. Ein parallel verarbeitendes Gerät zum gleichzeitigen Verarbeiten einer Vielzahl von Befehlen mit einer Vielzahl funktioneller Einheiten (4, 5, 6, 7) zum Ausführen entsprechender vorbestimmter Funktionen, einem Befehlsspeicher (1) zum Speichern von Befehlen, einer Befehlsabruf-Einheit (2) zum Abrufen einer Vielzahl von Befehlen aus dem Befehlsspeicher, einer Decodier-Einrichtung (3) zum Erkennen der Befehle unter den abgerufenen Befehlen, die gleichzeitig ausgeführt werden können, um diese Befehle, falls vorhanden, an entsprechende funktionelle Einheiten zuzuführen, wobei die Decodier-Einrichtung (3) eine Einrichtung (10), mit der bestimmt wird, ob ein Sprungbefehl in der Vielzahl der abgerufenen Befehle enthalten ist und eine Einrichtung (10), zum Zuführen eines eine Sprungerzeugung anzeigenden Signales (taken, not taken), das anzeigt, ob ein Sprung gemäß dem Sprungbefehl erzeugt ist, enthält, wobei jede der funktionellen Einheiten eine Befehls-Ausführungs-Einheit (EX) zum Ausführen eines Befehls aus der Decodier-Einrichtung (3) und eine Maschinenzustands-Änderungs-Einheit (WB) enthält, um gemäß dem Ausführungsergebnis der Befehlsausführungs-Einheit einen entsprechenden Maschinenzustand zu ändern, einer Einrichtung (10), um einer entsprechenden funktionellen Einheit einen Befehl zuzuführen, dem eine Gültigkeits-Kennung, die die Gültigkeit/Ungültigkeit des der funktionellen Einheit zugeführten Befehles anzeigt, und eine Schreibverzögerungs-Kennung zugefügt wurde, die anzeigt, ob der Befehl nach einem Sprungbefehl kommt, und einer Steuerungs-Einrichtung (100), um als Antwort auf die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung eine Operation der Maschinenzustandsänderungs-Einheit in einen der Zustände, den das Ausführungsergebnis haltenden Zustand, den das Ausführungsergebnis ungültigmachenden Zustand, und den Maschinenzustandsänderungs-Ausführungszustand, zu setzen.

2. Ein parallel verarbeitendes Gerät gemäß Anspruch 1, dadurch gekennzeichnet, daß die Steuerungseinrichtung eine Einrichtung (29) enthält, um als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung, die dem Befehl in der Maschinenzustands-Änderungseinheit zugefügt wurde, und auf eine Anzeige des eine Sprungerzeugung anzeigenden Signals, daß ein Sprung erzeugt ist, einen Sprung in der Maschinenzustands-Änderungseinheit ungültig zu machen, um die Änderung eines Maschinenzustandes gemäß dem Befehl zu unterbinden.

3. Ein parallel verarbeitendes Gerät gemäß Anspruch 1, dadurch gekennzeichnet, daß die Steuerungseinrichtung eine Einrichtung (29) enthält, um als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung, die dem Befehl in der Maschinenzustandsänderungs-Einheit zugefügt wurde, das der Befehl nach einem Sprungbefehl kommt, auf eine Anzeige des einen Sprungbefehl anzeigenden Signals, daß kein Sprung erzeugt ist, und auf einen Anzeige der dem Befehl zugefügten Gültigkeits-Kennung, daß der Befehl gültig ist, eine Änderung eines entsprechenden Zustandes, gemäß dem Befehl in der Maschinenzustandsänderungs-Einheit eine entsprechende Maschinenzustandsänderung zu aktivieren.

4. Ein parallel verarbeitendes Gerät gemäß Anspruch 1, dadurch gekennzeichnet, daß die Steuerungseinrichtung eine Einrichtung (29, 32) enthält, mit der der Wert der hinzugefügten Schreibverzögerungs-Kennung, als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung, die dem Befehl in der Maschinenzustandsänderungs-Einheit hinzugefügt wurde, daß ein Befehl nach einem Befehl kommt, und auf eine Anzeige des eine Sprungerzeugung anzeigenden Signals, daß noch nicht bestimmt ist, ob ein Sprung erzeugt wird, festgehalten wird.

5. Ein parallel verarbeitendes Gerät nach Anspruch 1, dadurch gekennzeichnet, daß die Steuerungseinrichtung eine Einrichtung (29, 30, 31, 32, 33) enthält, um als Antwort auf die Anzeige, der Schreibverzögerungs-Kennung, die den Befehl der Maschinenzustandsänderungs-Einheit zugefügt wurde, und auf die Anzeige des eine Sprungerzeugung anzeigenden Signals, daß noch nicht bestimmt ist, ob ein Sprung erzeugt wird, einen Befehl und die Schreibverzögerungs-Kennung und die dem Befehl hinzugefügte Gültigkeits-Kennung zu halten.

6. Ein parallel verarbeitendes Gerät nach Anspruch 1, dadurch gekennzeichnet, daß die Steuerungseinrichtung eine Einrichtung (29) enthält, um als Antwort auf die Anzeige der dem Befehl hinzugefügten Gültigkeits-Kennung, daß der Befehl ungültig ist, die Ausführung der Änderung eines Maschinenzustandes gemäß einem Befehl in der Maschinenzustandsänderungs-Einrichtung zu unterbinden.

7. Ein parallel verarbeitendes Gerät nach Anspruch 1, dadurch gekennzeichnet, daß die Steuerungseinrichtung eine Einrichtung (34) zum Decodieren eines Befehls in der Maschinenzustandsänderungseinrichtung, um ein Maschinenzustandsänderungsbefehlssignal gemäß dem Ergebnis der Decodierung zu erzeugen, eine Einrichtung (29), um als Antwort auf das Maschinenzustandsänderungsbefehlssignal, auf die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung, die dem Befehl in der Maschinenzustandsänderungs-Einheit hinzugefügt wird und auf eine eine Sprungerzeugung anzeigende Information ein eine Verzögerung anzeigendes Signal, daß den Zustand der Ausführungsverzögerung des Befehls in der Maschinenzustandsänderungseinheit anzeigt, und ein eine Befehlsgültigkeit anzeigendes Signal zu erzeugen, daß die Ungültigkeit des Befehls in der Maschinenzustandsänderungseinheit anzeigen, und eine Einrichtung (30, 32, 33) enthält, um als Antwort auf das eine Verzögerung anzeigende Signal und

das eine Befehlsgültigkeit anzeigende Signal, in der Maschinenzustandsänderungs-Einrichtung den Befehl und die Schreibverzögerungs-Kennung, gegen einen Befehl und eine Schreibverzögerungs-Kennung in der Ausführungs-Einheit zu ersetzen.

8. Ein parallel verarbeitendes Gerät nach Anspruch 7, dadurch gekennzeichnet, daß die Decodiereinrichtung eine Einrichtung enthält, mit der ein Ausführungs/Befehls-Ungültigkeitssignal erzeugt wird, um einen Befehl und ein Ausführungsergebnis in der Ausführungs-Einheit ungültig zu machen, wenn ein Sprung gemäß einem Sprungbefehl erzeugt wird, und

daß die Steuerungs-Einrichtung eine Einrichtung (31) enthält, mit der als Antwort auf das Ausführungs/Befehls-Ungültigkeitssignal, auf das eine Verzögerung anzeigende Signal und auf das eine Befehlsgültigkeit anzeigende Signal, die Gültigkeits-Kennung in der Maschinenzustandsänderungs-Einheit gegen eine dem Befehl in der Ausführungs-Einheit zugefügte Gültigkeits-Kennung zu ersetzen.

9. Ein parallel verarbeitendes Gerät gemäß Anspruch 1, dadurch gekennzeichnet, daß die Decodiereinrichtung eine Einrichtung (10) enthält, um ein eine Ausführung/einen Befehl anzeigendes Signal zu erzeugen, um als Antwort auf die Erzeugung eines Sprunges gemäß eines Sprungbefehls, ein Befehl in der Ausführungs-Einheit ungültig zu machen und daß das parallel verarbeitende Gerät weiterhin eine Einrichtung (42) enthält, um als Antwort auf das eine Ausführung/einen Befehl anzeigende Signal, eine Gültigkeits-Kennung, die den Befehl in der Ausführungs-Einheit in einem Ungültigkeits-Zustand zugefügt wird, zu setzen.

10. Ein parallel verarbeitendes Gerät nach Anspruch 1, dadurch gekennzeichnet, daß die Steuerungs-Einrichtung eine Einrichtung (29) enthält, die ein eine Verzögerung anzeigendes Signal erzeugt, daß eine Ausführungsverzögerung eines Befehls in der Maschinenzustandsänderungs-Einrichtung gemäß dem eine Sprungerzeugung anzeigenden Signals, dem Befehl in der Maschinenzustandsänderungs-Einheit und der Gültigkeits-Kennung und der dem Befehl zugefügten Schreibverzögerungs-Kennung ausführt, und

daß das parallel verarbeitende Gerät weiterhin eine Einrichtung (41, 42, 43, 44, 45) enthält, mit der als Antwort auf das eine Verzögerung anzeigende Signal, ein Befehl, eine Gültigkeits-Kennung und eine Schreibverzögerungs-Kennung in der Ausführungs-Einheit festgehalten wird.

11. Ein parallel verarbeitendes Gerät mit einem Befehlsspeicher (1), einer Abruf-Einheit (2), mit der gleichzeitig eine Vielzahl von Befehlen von dem Befehlsspeicher abgerufen werden, eine Vielzahl funktioneller Einheiten (4, 5, 6, 7), mit denen entsprechende vorbestimmte Funktionen ausgeführt werden, ein Decodier-Einheit (3) mit der die durch die Abruf-Einheit abgerufenen Befehle decodiert werden, um ein Befehl, der ausgeführt werden kann an eine entsprechende funktionelle Einheit abzugeben, wobei jede der funktionellen Einheiten eine Ausführungs-Einheit (EX) zum Ausführen eines empfangenen Befehls und einer Maschinenzustandsänderungs-Einheit (WB) enthält, um gemäß einem Ausführungsergebnis der Ausführungs-Einheit einen entsprechenden Maschinenzustand zu

ändern, wobei die Abruf-Einheit, die Decodier-Einheit und die Ausführungs-Einheit und die Maschinenzustandsänderungs-Einheit eine parallel verarbeitende Operation durchführen, wobei das Gerät eine Decodier-Einrichtung (10), die in der Decodier-Einheit vorgesehen ist, um einen Sprungbefehl unter den von der Abruf-Einheit abgerufenen Befehlen zu erkennen, um ein eine Sprungerzeugung anzeigendes Signal (taken, not taken) zu erzeugen, das anzeigt, ob ein Sprung gemäß dem Sprungbefehl erzeugt wird, um eine Gültigkeits-Kennung zu erzeugen, um die Gültigkeit oder Ungültigkeit des an eine entsprechende funktionelle Einheit abgegebenen Befehls anzuzeigen, und um eine Schreibverzögerung-Kennung zu erzeugen, die anzeigt, ob ein abgegebener Befehl ein Befehl nach einem Sprungbefehl ist, wobei die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung mit dem abgegebenen Befehl verbunden sind und an die Ausführungs-Einheit der entsprechend funktionellen Einheit übermittelt werden,

einen ersten Zwischenspeicher (11, 12, 13, 14, 15), der in der Ausführungs-Einheit vorgesehen ist, um den Befehl, die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung, die von der Decodiereinrichtung empfangen wurden, als Antwort auf erstes Zeittaktsignal zwischenzuspeichern, einen zweiten Zwischenspeicher (17, 18, 19, 20, 35, 36), der in der Ausführungs-Einheit vorgesehen ist, um eine Ausgabe des ersten Zwischenspeichers als Antwort auf ein zweites Zeittaktsignal (L) zwischenzuspeichern, wobei der zweite Zwischenspeicher eine Ausführungsergebnis-Zwischenspeichereinrichtung (20) enthält, die ein Ausführungsergebnis der vorbestimmten Funktion der entsprechenden funktionellen Einheit zwischenspeichert, einen dritten Zwischenspeicher (21, 22, 23, 24), der in der Maschinenzustandsänderungs-Einrichtung vorgesehen ist, um die Ausgabe der zweiten Zwischenspeicher als Antwort auf das erste Zeittaktsignal zwischenzuspeichern, einen vierten Zwischenspeicher (25, 26, 27, 28), der in der Maschinenzustandsänderungs-Einrichtung vorgesehen ist, um eine Ausgabe der dritten Zwischenspeicher als Antwort auf das zweite Zeittaktsignal zwischenzuspeichern, eine Rückschreibe-Decodiereinrichtung (34), um einen in der dritten Zwischenspeicher zwischengespeicherten Befehl zu decodieren, wobei die Rückschreib-Decodiereinrichtung ein Maschinenzustandsänderungs-Befehlssignal (w RF) erzeugt, wenn der decodierte Befehl eine Maschinenzustandsänderung anzeigt, und eine Steuerungs-Einrichtung (100) aufweist, die durch die Wahl eines der Ausführungszustände des mit der Rückschreibe-Decodiereinrichtung decodierten Befehls, die Maschinenzustandsänderungs-Einrichtung in einen der Zustände des Maschinenzustandsänderungs-Ausführens, des Ausführungsungültigmachens und des Ausführungsverchiebens, setzt, die Ausführung darin verschiebt, und als Antwort auf das Maschinenzustandsänderungs-Befehlssignal, auf das eine Sprungerzeugung anzeigende Signal und auf die Gültigkeits-Kennung und auf die Schreibverzögerungs-Kennung, den in der Zwischenspeicher zwischengespeicherten decodierten Befehl ungültig macht.

12. Ein parallel verarbeitendes Gerät nach An-

spruch 11, dadurch gekennzeichnet, daß die Steuerungs-Einrichtung eine Einrichtung (29) zum Erzeugen eines eine Verzögerung anzeigendes Signales (WB busy), das als Antwort auf das Maschinenzustandsänderungsbefehls-Signal, auf das eine Sprungerzeugung anzeigende Signal und auf die Gültigkeits-Kennung und auf die Schreibverzögerungs-Kennung, die in den vier Zwischenspeichereinrichtungen gespeichert sind, anzeigt, daß die Ausführung der Änderung durch die Maschinenzustandsänderungs-Einrichtung sich verzögert, und eine Einrichtung (30, 32, 33) enthält, um als Antwort auf das eine Verzögerung anzeigende Signal, selektiven entweder des Befehls und der Schreibverzögerungs-Kennung aus der zweiten Zwischenspeichereinrichtung oder des Befehls und der Schreibverzögerungs-Kennung in der vierten Zwischenspeichereinrichtung an die dritte Zwischenspeichereinrichtung, zu übermitteln.

13. Ein parallel verarbeitendes Gerät nach Anspruch 12, dadurch gekennzeichnet, daß die Decodiereinrichtung in der Decodier-Einheit ein eine Ausführung löschendes Signal (EX can) erzeugt, das das Löschen eines Befehls in der Ausführungseinheit bei der Erzeugung eines Sprunges gemäß dem Sprungbefehl anzeigt, und daß die Steuerungseinrichtung weiterhin eine Einrichtung (29) zum Erzeugen eines eine Rückschreiblöschung anzeigendes Signal (WB can) erzeugt, daß das Löschen der Ausführung eines Befehls in der Maschinenzustandsänderungs-Einrichtung als Antwort auf das Maschinenzustandsänderungsbefehls-Signals, auf das eine Sprungerzeugung anzeigende Signal und auf die Gültigkeits-Kennung und auf die Schreibverzögerungs-Kennung, die in der vierten Zwischenspeichereinrichtung gespeichert ist anzeigt, und eine Einrichtung (31) beinhaltet, um als Antwort auf das eine Ausführungslöschung anzeigende Signal, auf das eine Verschiebung anzeigende Signal und auf das eine Rückschreiblöschung anzeigende Signal, selektiv eine der Kennungen, die in der zweiten Zwischenspeichereinrichtung gespeicherte Gültigkeits-Kennung, die in der vierten Zwischenspeichereinrichtung gespeicherte Gültigkeits-Kennung, und die eine Ungültig anzeigende Kennung eines vorbestimmten logischen Niveaus, an die dritte Zwischenspeichereinrichtung zuzuführen.

14. Ein parallel verarbeitendes Gerät nach Anspruch 12, dadurch gekennzeichnet, daß weiterhin eine Einrichtung (41, 42, 43, 44, 45) enthalten ist, mit der als Antwort auf das eine Verzögerung anzeigende Signal, entweder ein Befehl, eine Gültigkeits-Kennung und eine Schreibverzögerungs-Kennung aus der Decodier-Einrichtung oder der Befehl, die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung, die in der zweiten Zwischenspeichereinrichtung gespeichert ist, zu der ersten Zwischenspeichereinrichtung passieren.

15. Ein parallel verarbeitendes Gerät nach Anspruch 13, dadurch gekennzeichnet, daß weiterhin eine Einrichtung zum Setzen der Gültigkeits-Kennung, die in der zweiten Zwischenspeichereinrichtung als Antwort auf das Ausführungslöschungs-Signal gespeichert ist, enthalten ist.

16. Ein Verfahren zum Ungültigmachen eines Befehls in einer parallel verarbeitenden Einheit, in der

eine Vielzahl von Befehlen aus dem Befehlsspeicher (1) abgerufen werden, um die abgerufenen Befehle durch eine Vielzahl funktioneller Einheit (4, 5, 6, 7) parallel zu verarbeiten, wobei durch einen empfangenen Befehl jede eine vorbestimmte Funktion durchführt, und die eine Befehlsausführungseinheit (EX) zum Ausführen eines empfangenen Befehls und eine Maschinenzustandsänderungseinheit (WB) enthält, um den Zustand einer entsprechenden Maschine (9) gemäß einem Ausführungsergebnis der Befehlsausführungseinheit zu ändern, gekennzeichnet durch die Verfahrensschritte, daß die parallel zu verarbeitenden Befehle unter den abgerufenen Befehlen erkannt werden, und daß diese ausführbaren Befehle entsprechenden funktionellen Einheiten zugeführt werden, daß der Erkennungsschritt die Schritte enthält, daß erkannt wird, ob ein Sprungbefehl in der abgerufenen Vielzahl von Befehlen enthalten ist und daß ein eine Sprungerzeugung anzeigendes Signales (taken/not taken) erzeugt wird, das anzeigt, ob ein Sprung gemäß dem Sprungbefehl erzeugt ist, daß das Verfahren weiterhin die Schritte enthält, daß eine Gültigkeits-Kennung, die eine Gültigkeit und eine Ungültigkeit eines an eine entsprechende funktionelle Einheit zugeführten Befehles, und daß eine Schreibverzögerungs-Kennung, die anzeigt, ob der Befehl ein Befehl nach einem Sprungbefehl gemeinsam in der entsprechenden funktionellen Einheit ist, zugeführt werden, und daß die Operation der Maschinenzustandsänderungseinheit als Antwort auf die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung in einen der Zustände, den ein Ausführungsergebnis haltenden Zustand, den ein Ausführungsergebnis gültigmachenden Zustand und den eine Maschinenzustandsänderung ausführenden Zustand, gesetzt wird.

17. Verfahren gemäß Anspruch 16, dadurch gekennzeichnet, daß der Schritt enthalten ist, daß als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung, die in der Maschinenzustandsänderungseinrichtung dem Befehl hinzugefügt ist, daß der Befehl nach einem Sprungbefehl kommt und auf eine Anzeige des einer Sprungerzeugung anzeigende Signales, daß ein Sprung erzeugt ist, ein Befehl in der Maschinenzustandsänderungseinheit ungültig gemacht wird und die Änderung der Maschinenzustandsänderung gemäß dem Befehl unterbunden wird.

18. Verfahren gemäß Anspruch 16, dadurch gekennzeichnet, daß der Schritt des Setzens den Schritt enthält, daß als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung die dem Befehl in der Maschinenzustandsänderungseinrichtung hinzugefügt ist, daß der Befehl nach dem Sprungbefehl kommt, auf eine Anzeige des einer Sprungerzeugung anzeigenden Signales, daß kein Sprung erzeugt ist, und auf eine Anzeige der Gültigkeits-Kennung, die dem Befehl hinzugefügt ist, daß der Befehl gültig ist, gemäß einem Befehl in der Maschinenzustandsänderungseinrichtung die Änderung des entsprechenden Maschinenzustands aktiviert wird.

19. Verfahren nach Anspruch 16, dadurch gekennzeichnet, daß der Schritt des Setzens den Schritt

enthält, das als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung, die dem Befehl in der Maschinenzustandsänderungs-Einheit hinzugefügt ist, daß ein Befehl nach einem Sprungbefehl kommt und auf die Anzeige des eine Sprungerzeugung anzeigenden Signals, daß noch nicht bestimmt ist ob ein Sprung erzeugt wird, der Wert der Schreibverzögerungs-Kennung festgehalten wird.

20. Ein Verfahren nach Anspruch 16, dadurch gekennzeichnet, daß der Schritt des Setzens den Schritt enthält, daß als Antwort auf eine Anzeige der Schreibverzögerungs-Kennung, die den Befehl in der Maschinenzustandsänderungs-Einheit hinzugefügt ist, daß der Befehl nach dem Sprungbefehl kommt und auf eine Anzeige, des eine Sprungerzeugung anzeigenden Signals, daß noch nicht bestimmt ist, ob ein Sprung erzeugt wird, ein Befehl, eine Schreibverzögerungs-Kennung und ein dem Befehl hinzugefügte Gültigkeits-Kennung festgehalten werden.

21. Ein Durchführungsvorgang gemäß Anspruch 16, dadurch gekennzeichnet, daß der Schritt des Setzens einen Schritt enthält, daß als Antwort auf eine Anzeige der Gültigkeits-Kennung, die dem Befehl in der Maschinenzustandsänderungs-Einheit hinzugefügt ist, daß der Befehl ungültig ist, die Ausführung der Maschinenzustandsänderung gemäß dem Befehl unterbunden wird.

22. Verfahrensdurchführung gemäß Anspruch 16, dadurch gekennzeichnet, daß der Schritt des Setzens die Schritte enthält, daß ein Befehl in der Maschinenzustandsänderungs-Einheit decodiert wird und ein Maschinenzustandsänderungsbefehls-Signal gemäß dem Ergebnis der Decodierung erzeugt wird, daß als Antwort auf das Maschinenzustandsänderungsbefehls-Signal, auf die Gültigkeits-Kennung und auf die Schreibverzögerungs-Kennung, die dem Befehl in der Maschinenzustandsänderungs-Einheit hinzugefügt ist, und auf die eine Sprungerzeugung anzeigende Information, ein eine Verzögerung anzeigendes Signal, daß den Zustand der verzögerten Ausführung des Befehls in der Maschinenzustandsänderungs-Einheit anzeigt, und ein eine Befehlsgültigkeit anzeigendes Signal erzeugt wird, daß die Ungültigkeit des Befehls in der Maschinenzustandsänderungs-Einheit anzeigt, und daß für den Befehl und die Schreibverzögerungs-Kennung in der Maschinenzustandsänderungs-Einheit als Antwort auf das eine Verzögerung anzeigende Signal und auf das eine Gültigkeit anzeigende Signal, ein Befehl und eine Schreibverzögerungs-Kennung in der Ausführung-Einheit eingefügt wird.

23. Ein Verfahren nach Anspruch 22, dadurch gekennzeichnet, daß weiterhin der Schritt enthalten ist, daß ein Ausführungs/Befehls-Ungültigkeitssignal erzeugt wird, um einen Befehl und ein Ausführungsergebnis in der Ausführung-Einheit ungültig zu machen, wenn ein Sprung gemäß dem Sprungbefehl erzeugt wird und daß der Schritt des Setzens den Schritt enthält, daß als Antwort auf das Ausführungs/Befehls-Ungültigkeitssignal, auf das eine Verzögerung anzeigende Signal und auf das eine Befehlsgültigkeit anzeigende Signal, die Gültigkeits-Kennung, die dem Befehl in der Ausführung-Einheit hinzugefügt ist,

für die Gültigkeits-Kennung in der Maschinenzustandsänderungs-Einheit eingesetzt wird.

24. Ein Verfahren gemäß Anspruch 16, dadurch gekennzeichnet, daß weiterhin die Schritte enthalten sind, daß ein Ausführungs/Befehls-Ungültigkeitssignal erzeugt wird, um einen Befehl in der Ausführung-Einheit, als Antwort auf die Erzeugung eines Sprunges gemäß dem Sprungbefehl, zu erzeugen, und

daß eine Gültigkeits-Kennung, die dem Befehl in der Ausführung-Einheit hinzugefügt ist, als Antwort auf das Ausführungs/Befehls-Ungültigkeitssignal in den Ungültigkeitszustand gesetzt wird.

25. Ein Verfahren nach Anspruch 16, dadurch gekennzeichnet, daß weiterhin die Schritte enthalten sind,

daß gemäß dem eine Sprungerzeugung anzeigenden Signal, dem Befehl in der Maschinenzustandsänderungs-Einheit und der Gültigkeits-Kennung und der Schreibverzögerungs-Kennung, die dem Befehl hinzugefügt ist, ein eine Verzögerung anzeigendes Signal erzeugt wird, um eine Ausführungsverzögerung eines Befehls in der Maschinenzustandsänderungs-Einheit anzuzeigen, und daß als Antwort auf das eine Verzögerung anzeigende Signal der Befehl, die Gültigkeits-Kennung und die Schreibverzögerungs-Kennung in der Ausführung-Einheit festgehalten werden.

26. Ein parallel verarbeitendes Gerät mit einer Befehlsspeichereinrichtung (1) zum Speichern von Befehlen in einer vorbestimmten Verarbeitungsfolge, eine Speichereinrichtung (9) zum Speichern von Ergebnissen von ausgeführten Befehlen, eine Befehlsabruf-Einrichtung (2) zum Abrufen einer Vielzahl zu bearbeitender Befehle aus der Befehlsspeichereinrichtung, eine Vielzahl funktioneller Einheiten (4, 5, 6, 7, 10) zur Befehlsverarbeitung, eine Decodiereinrichtung (3, 10) mit der gleichzeitig die Befehle unter den von der Befehlsabruf-Einrichtung abgerufenen Befehlen, die parallel ausführbar sind, an die Vielzahl der entsprechenden funktionellen Einheiten zugeführt werden, wobei die Vielzahl der funktionellen Einheiten eine Ausführung-Einrichtung (EX), mit der als Antwort auf gleichzeitig empfangene Befehle, vorbestimmte Funktion gleichzeitig ausgeführt werden und die eine Schreibeinrichtung (WB; 28, 19, 94) enthält mit der die Ausführungsergebnisse der funktionellen Einheiten in die Speichereinrichtung geschrieben werden,

eine Verzögerungs-Einrichtung (10, 100), die als Antwort auf die gleichzeitig empfangenen Befehle das Schreiben durch die Schreibeinrichtung von wenigstens einem Ausführungsergebnis verzögert, eine Gültigkeitseinrichtung (10), die als Antwort auf die Ausführungsergebnisse einer anderen der funktionellen Einheiten, ein Gültigkeits-Signal (not taken) erzeugt, daß anzeigt, daß Befehle, die den verzögerten Ausführungsergebnissen entsprechen ungültig sind, und einer Steuerungs-Einrichtung (100) die als Antwort auf das Gültigkeits-Signal ein Steuersignal der Schreibeinrichtung zuführt, um die Verzögerung des Schreibens in die Speichereinrichtung von wenigstens einem der Ausführungsergebnisse zu beenden.

27. Parallel verarbeitendes Gerät gemäß Anspruch

26, dadurch gekennzeichnet, daß das Ergebnis einer anderen funktionellen Einheiten einem Sprungbefehl entspricht, daß wenigstens eines der Ausführungsergebnisse den Befehlen entspricht, die dem Sprungbefehl in der Verarbeitungsfolge untergeordnet sind, und daß das Gültigkeits-Signal anzeigt, das ein Sprung durch den Sprungbefehl nicht genommen ist, und daß das Gerät weiterhin eine Löscheinrichtung (10), um als Antwort auf ein Ausführungsergebnis das anzeigt, daß der Sprung genommen ist, eine Löschungssignal (taken) zu erzeugen, und eine Löscheinrichtung (100, 29) enthält, die verzögerte Ausführungsergebnisse löscht, ohne als Antwort auf das Löschungssignal, in die Speichereinrichtung zu schreiben.

28. Verfahren zur Parallelverarbeitung von Befehlen in parallel verarbeitenden Geräten mit einer Befehlsspeichereinrichtung (1) zum Speichern von Befehlen in einer vorbestimmten Verarbeitungsfolge, einem Datenspeicher (19) und einer Vielzahl von funktionellen Ausführungseinheiten (4, 5, 6, 7), dadurch gekennzeichnet, daß das Verfahren die Schritte enthält, daß eine vorbestimmte Anzahl von Befehlen abgerufen werden, die einen von einem Sprungbefehl abhängigen Befehl enthalten daß in den Datenspeicher ein Ergebnis des ausgeführten Befehls, der nicht von einem Sprungbefehl abhängt, geschrieben wird, daß in den Datenspeicher ein Ergebnis des ausgeführten Befehls, der von einem Sprungbefehl abhängt, verzögert geschrieben wird, daß der Verzögerungsschritt als Antwort auf die Ausführung des abhängigen Sprungbefehls beendet wird, wenn ein Sprung nicht ausgeführt wird, und daß die von dem Sprungbefehl abhängigen Befehle gelöscht werden, ohne das deren Ausführungsergebnisse geschrieben werden, falls ein Sprung auszuführen ist.

29. Ein parallel verarbeitendes Gerät mit einer Befehlsdecoder (10), einer Vielzahl funktioneller Einheiten (4, 5, 6, 7) zur Befehlsverarbeitung, einer Datenspeichereinrichtung (9), und eine Steuerungs-Einrichtung (100), dadurch gekennzeichnet, daß der Befehlsdecoder eine Befehlszuführungseinrichtung, um konkurrierend eine Vielzahl von Befehlen an die Vielzahl von funktionellen Einheiten zuzuführen, um die Befehle gleichzeitig auszuführen und eine Kennungserzeugungs-Einrichtung enthält, um wenigstens ein Kennungs-Signal zuzuführen, daß, an jeden Befehl angehängt, abgegeben wird, und daß die Steuerungs-Einrichtung eine Schreibereinrichtung (29) enthält, um als Antwort auf wenigstens ein Kennungs-Signal, die Ergebnisse der von den funktionellen Einheiten ausgeführten Befehlen, selektiv in die Datenspeichereinrichtung zu schreiben, daß wenigstens ein Kennungs-Signal anzeigt, ob ein Ausführungsergebnis eines zugeordneten Befehls in die Datenspeichereinrichtung zu schreiben ist oder nicht.

30. Ein parallel verarbeitendes Gerät nach An-

spruch 29, dadurch gekennzeichnet, daß wenigstens ein Kennungs-Signal ein Verzögerungsbefehls-Signal enthält, das anzeigt, daß ein zugeordneter Befehl in der Verarbeitungsfolge nach einem Sprungbefehl kommt.

31. Ein Verfahren zur parallelen Befehlsverarbeitung mit den Schritten, daß eine Vielzahl von Befehlen decodiert werden, die gleichzeitig aus dem Befehlsspeicher (1) abgerufen wurden, um die gleichzeitig ausführbaren Befehle zu erkennen, daß wenigstens ein Kennungs-Signal erzeugt wird, das an jeden der abgerufenen Befehle angehängt wird, wobei das Kennungs-Signal anzeigt, daß ein Ausführungsergebnis eines zugeordneten Befehls die Änderung eines entsprechenden Maschinenzustands verursacht, daß die ausführbaren Befehle gleichzeitig ausgeführt werden, und daß die Änderung des entsprechenden Maschinenzustandes, als Antwort auf jedes zugeordnete Kennungssignal, gemäß den Ausführungsergebnissen der ausführbaren Befehle verzögert wird.

32. Verfahren nach Anspruch 31, dadurch gekennzeichnet, daß wenigstens ein Kennungs-Signal anzeigt, daß ein zugeordnetes Signal in der Verarbeitungsfolge nach einem Sprungbefehl kommt.

Hierzu 9 Seite(n) Zeichnungen

FIG.1 Stand der Technik

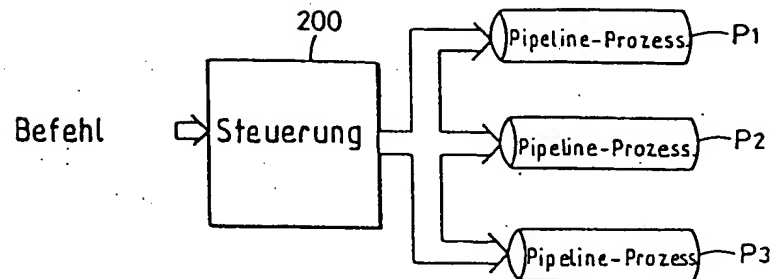


FIG.2

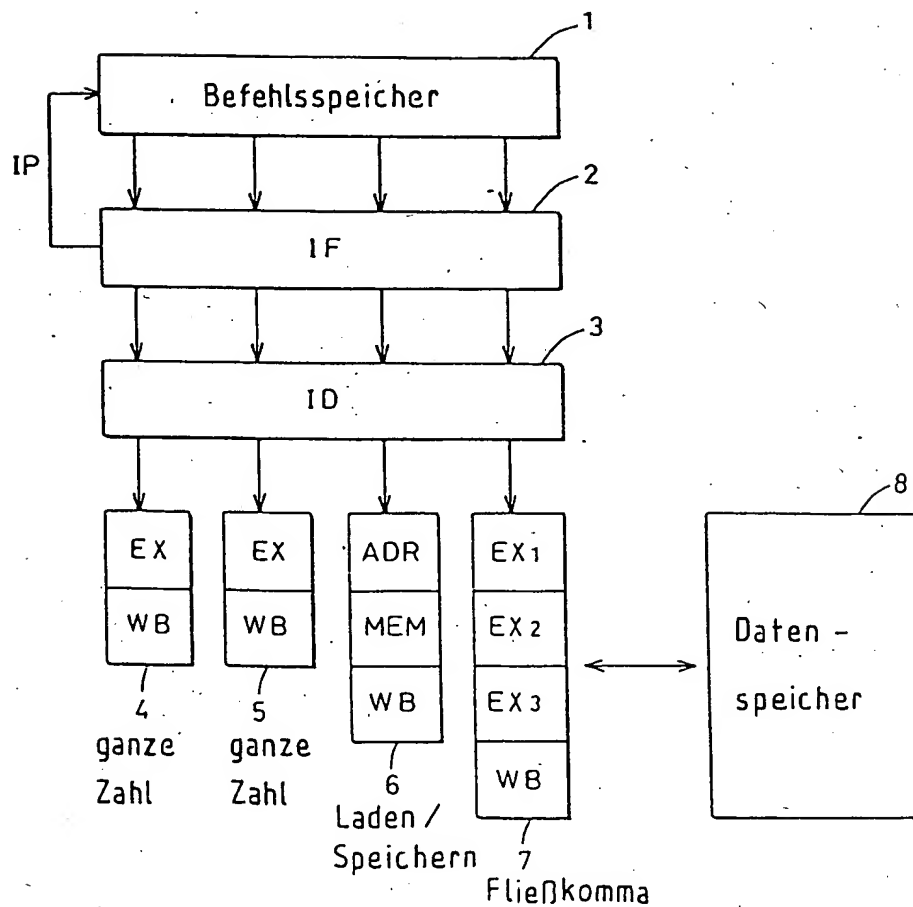


FIG. 3 Stand der Technik

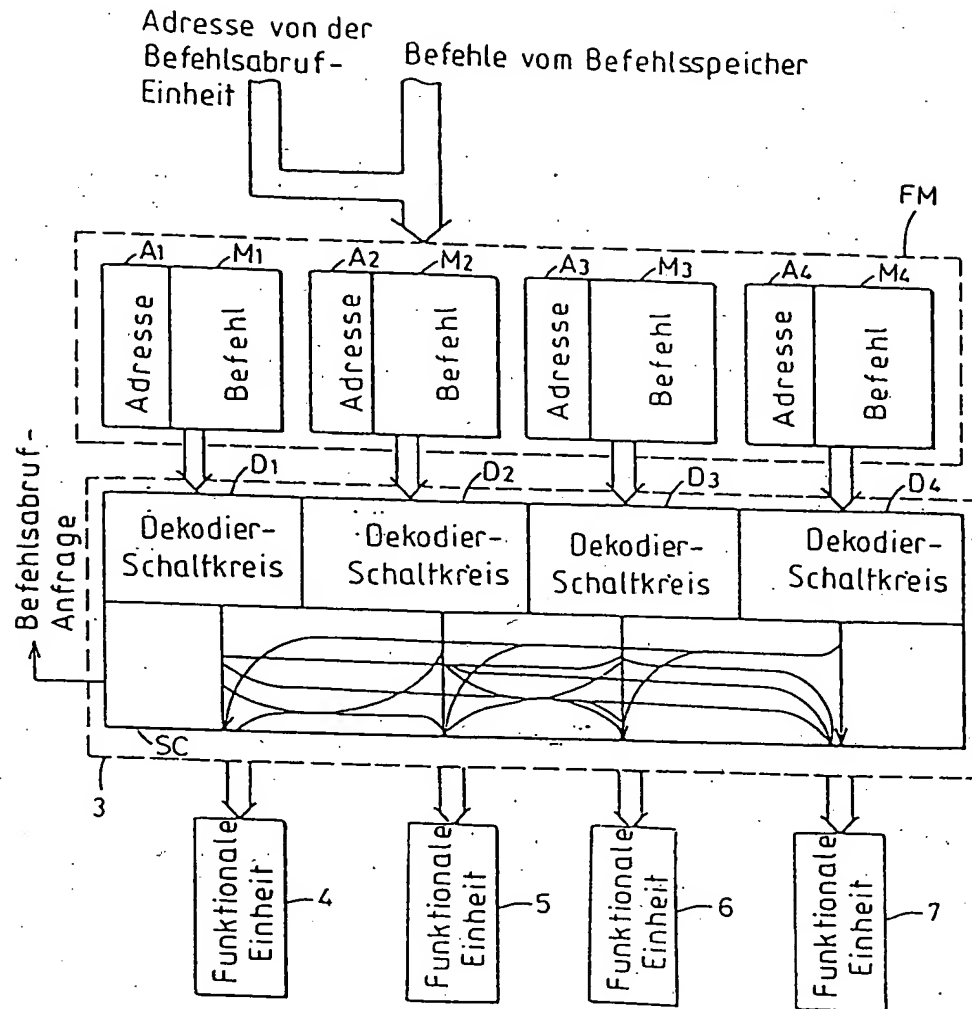


FIG. 4 Stand der Technik

Zyklus	Befehl				
1	1	2	3	4	: Abgabe von 1
2		2	3	4	: Abgabe von 2,3
3				4	: Abgabe von 4
4	5	6	7	8	: Abgabe von 5,6

FIG. 5 Stand der Technik

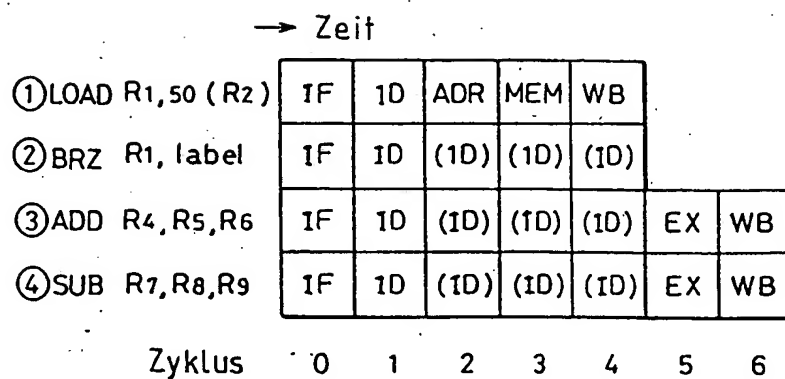


FIG. 6

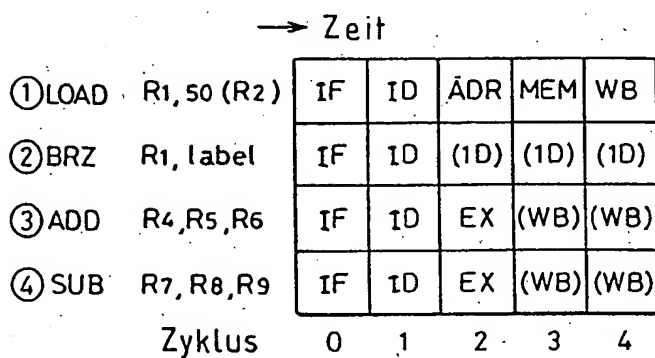


FIG. 8

Zwei-Phasen-Zeittakt

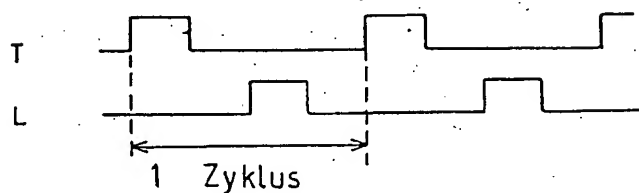


FIG. 7

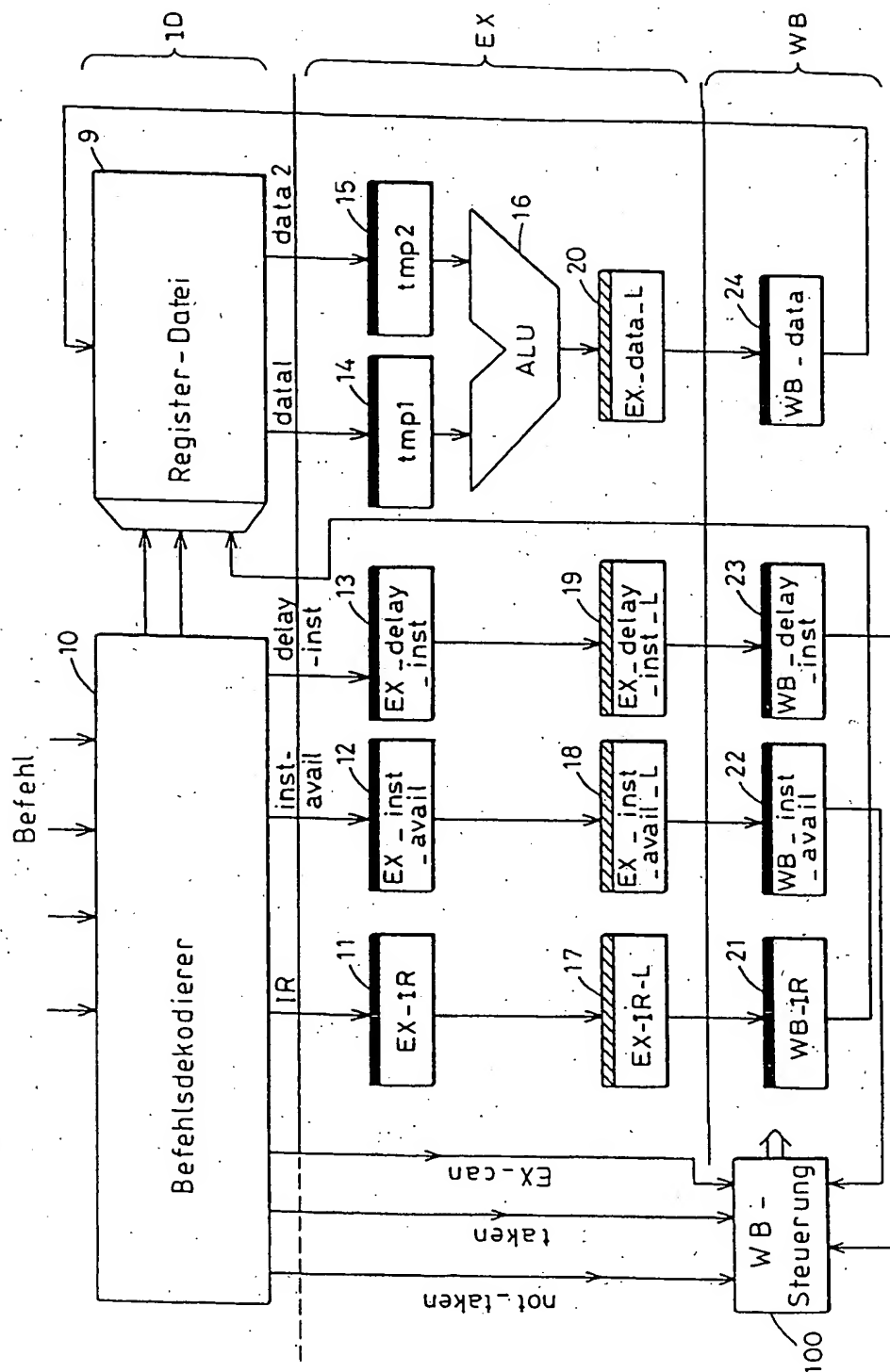


FIG.10

Logik der "WB_busy_logic"

WB _inst_avail _L	w_RF	WB_delay _inst_L	not_taken	taken	wen	WB_busy	WB_can
0	X	X	X	X	0	0	0
1	0	X	X	X	0	0	0
1	1	0	X	X	1	0	0
1	1	1	0	0	0	1	0
1	1	1	1	0	1	0	0
1	1	1	0	1	0	0	1

FIG.11

Logik der "in_logic 1"

WB_can	EX_can	WB_busy	WB_inst_avail
1	0	X	EX_inst_avail_L
1	1	X	0
0	X	0	EX_inst_avail_L
0	X	1	WB_inst_avail_L

FIG. 12

Logik der "in_logic 2"

WB _ busy	WB_IR	WB_delay_inst	WB _ data
1	WB_IR_L	WB_delay_inst_L	WB_data_L
0	EX_IR_L	EX_delay_inst_L	EX_data_L

FIG. 13

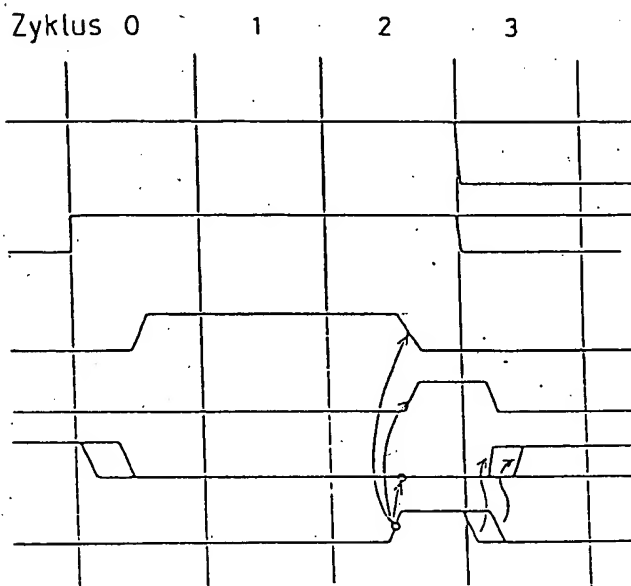


FIG. 14

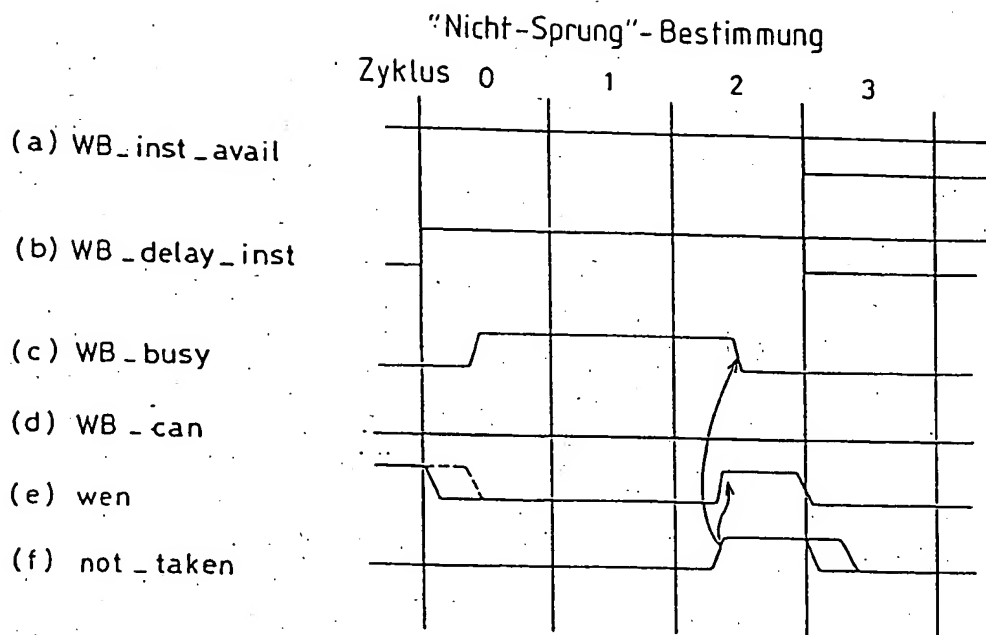


FIG. 16

Logik der "in_logic 3"

WB_busy	EX_IR	EX_delay_inst	EX_inst_avail	tmp 1 (2)
1	EX_IR - L	EX_delay_inst_L	EX_inst_avail_L	tmp 1(2) - L
0	IR	delay_inst	inst_avail	data 1 (2)

FIG. 17

EX_can	EX_inst_avail_L
0	EX_inst_avail
1	0

FIG. 15

